

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan dari pembahasan-pembahasan pada bab-bab sebelumnya, maka dapat ditarik kesimpulan dari Tugas Akhir ini :

1. Sistem Informasi Kantor Sistem Informasi (SIKSI) yang dibangun untuk Universitas Atma Jaya Yogyakarta telah berhasil dibuat dan berjalan dengan baik untuk dapat melakukan fungsi yang memudahkan KSI dalam melakukan *monitoring* pengembangan aplikasi.
2. Sistem informasi Kantor Sistem Informasi (SIKSI) yang dibangun dapat menyediakan pengelolaan *project* sistem informasi.

6.2 Saran

Beberapa saran dan masukan yang dapat disampaikan penulis terhadap pembuatan Sistem Informasi Kantor Sistem Informasi ini dimasa yang akan datang adalah :

1. Sistem ini diharapkan dapat melakukan pemantauan aplikasi sudah dikembangkan dan digunakan.
2. Sistem ini diharapkan dapat melakukan pemantauan terhadap sumber daya perangkat keras yang dimiliki oleh KSI contoh router, switch dan server.
3. Sistem ini diharapkan dapat memberikan notifikasi menggunakan email kepada staff IT ketika mendapatkan penugasan *project*.

4. Sistem ini diharapkan dapat menyediakan data *project* portofolio mengenai pembangunan sistem informasi.



DAFTAR PUSTAKA

- Avison, D.E., and fitzgerald, G., 2003, *Information Systems Development : Methodologies, Techniques and Tools*, 3rd ed, McGraw-Hill, London.
- Bates, T, 2005, *Technology, e-learning and distance education : second edition* , Routledge, New York.
- Boss, R. W., 2006, *Client-Server Technology*, American Library Association, pp.1-4.
- Burch, J., and Grudnitski, G., 1986, *Information Systems Theory and Practice*, John Wiley and Sons, New York.
- Gondodiyoto, Sanyoto, 2007, *Audit Sistem Informasi + Pendekatan CobIT. Edisi Revisi*. Mitra Wacana Media, Jakarta.
- Gondodiyoto, Sanyoto, dan Henny Hendarti, 2006, *Audit Sistem Informasi*. MitraWacana Media, Jakarta.
- Hartono, Jogiyanto, 1990, *Analisa dan Desain Sistem Informasi*, Andi, Yogyakarta.
- Hartono, Jogiyanto, 1999, *Analisis & desain : sistem informasi : pendekatan terstruktur teori dan praktek aplikasi bisnis*, Andi, Yogyakarta.
- Hartono, Jogiyanto, 2000, *Pengenalan Komputer : Dasar Ilmu Komputer, Pemograman, Sistem Informasi, dan Intelegensi Buatan Edisi 3. Cet.Kedua*, Andi, Yogyakarta.
- Jim, D., 2007, "What is asset management and where do you start?", American Water Works Association Journal, Vol. 99 No. 10, p.26.

- Jurison, J., September 1999, *Software Project Management: Manager's View*, Communication of the Association for Information Systems Vol. 2, Article 17.
- Kerzner, H., 1995, *Project management: A systems approach to planning, scheduling, and controlling*, 5th Edition, Van Nostrand Reinhold, New York.
- Laura, Lemay, 2001, *Desain Grafik dan Halaman Web*, Bandung : Informatika.
- Maffeis, S., 2005, Client-Server Computing, media.wiley.com/assets/152/06/computer.pdf, diakses tanggal 6 juni 2012.
- Mitchell, John S., 2006, *Physical Asset Management Handbook*, CLARION Technical, Boston.
- O'Brien, James A., 2005. *Introduction to Information System*, McGraw-Hill, Singapore, p8.
- Prawitasari, Yuliana, 2012, Sistem Informasi Manajemen Aset Tetap IT Telkom Dengan Menggunakan Bantuan GIS (Geographic Information System), Institut Teknologi Telkom, http://ppm.ittelkom.ac.id/index.php?option=com_content&view=article&id=161:sistem-informasi-manajemen-aset-tetap-it-telkom-dengan-menggunakan-bantuan-gis-geographic-information-system&catid=37:pdi-2011&Itemid=63, diakses tanggal 6 juni 2012.
- Schuetze W. P, September 1993, *Accounting Horizons* vol.7, No. 3, [immamu.edu.sa](http://www.immammu.edu.sa), <http://www.immammu.edu.sa/Data/abstract/management/aca/What%20Is%20An%20Asset.pdf>, diakses pada tanggal 3 februari 2012.
- Senn, J., 1989, *Analysis and design of information systems*, McGraw-Hill, New York.
- Sheriff, Paul D., 2011, Introduction to ASP.net and Web Forms, Microsoft, msdn.microsoft.com/en-us/library/ms973868.aspx#introwebforms_topic1, diakses tanggal 10 juni 2012

Sudrajat, Irawan, 2007. *Lifecycle Asset Managemen*, <http://assetmanagement.wordpress.com/2007/06/14/lifecycle-asset-management/>, diakses pada tanggal 3 februari 2012.

Utaminingsih, S., 2008, Pengembangan Sistem Informasi Manajemen Proyek di PT. Thames PAM JAYA Menggunakan PHP dan MySql, *Skripsi*, Universitas Gunadarma, Jakarta.

Watson, R. T., 2007, *Information Systems*, Jacobs Foundation, Zurich, pp.20-22.

Whitten, Jeffery L. Bentley, Lonnie D. Dittman, Kevin C., 2004, *Systems Analysis and Design Methods*, 6th edition, McGraw Hill.

SKPL

SPESIFIKASI KEBUTUHAN PERANGKAT LUNAK

SIKSI

(Sistem Informasi Kantor Sitem Informasi)

Untuk :


Universitas Atma Jaya Yogyakarta

Dipersiapkan oleh:

Jefri Trio Vember / 5532

Program Studi Teknik Informatika - Fakultas Teknologi Industri

Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika Fakultas Teknologi Industri	Nomor Dokumen		Halaman
		SKPL-SIKSI		1/66
		Revisi		

DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	
G	

INDEX TGL	-	A	B	C	D	E	F	G
Ditulis oleh	JTV							
Diperiksa oleh	YSP							
Disetujui oleh	IW							

Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi

Daftar Isi

1. PENDAHULUAN	7
1.1 Tujuan	7
1.2 Lingkup Masalah	7
1.3 Definisi, akronim, dan singkatan	8
1.4 Referensi	9
1.5 Deskripsi Umum (Overview)	9
2. DESKRIPSI KEBUTUHAN	10
2.1 Perspektif Produk	10
2.2 Fungsi Produk	11
2.3 Karakteristik Pengguna	19
2.4 Batasan-batasan	20
2.5 Asumsi dan Ketergantungan	20
3. KEBUTUHAN KHUSUS	21
3.1 Kebutuhan antarmuka eksternal	21
3.1.1 Antarmuka pemakai	21
3.1.2 Antarmuka perangkat keras	21
3.1.3 Antarmuka perangkat lunak	21
3.1.4 Antarmuka Komunikasi	22
3.2 Kebutuhan Fungsionalitas Perangkat Lunak ..	23
3.2.1 Use Case Diagram	23
4. SPESIFIKASI RINCI KEBUTUHAN	24
4.1 Spesifikasi Kebutuhan Fungsionalitas	24
4.1.1 Use case Specification: Login	24
4.1.2 Use case Specification: Mengajukan Permohonan Pembuatan Aplikasi	25
4.1.3 Use case Specification: Mengajukan Penambahan Fitur	26
4.1.4 Use case Specification: Melakukan Manajemen Project	28
4.1.5 Use case Specification: Mengelola Fitur Project	30
4.1.6 Use case Specification: Mengelola Form Aplikasi	33
4.1.7 Use case Specification: Mengelola Report Aplikasi	35
4.1.8 Use case Specification: Mengelola Desain Aplikasi	37
4.1.9 Use case Specification: Melakukan Maintenance Aplikasi	39

4.1.10	Use case Specification: Monitoring Resource.....	40
4.1.11	Use case Specification: Mencetak Laporan Project Manager.....	41
4.1.12	Use case Specification: Melihat Laporan Perkembangan Project.....	42
4.1.13	Use case Specification: Menugaskan Project.....	43
4.1.14	Use case specification: Melihat Task	44
4.1.15	Use case specification: Mencetak Surat CCP	45
4.1.16	Use case specification: Melihat Laporan Sistem Informasi	46
4.1.17	Use case specification: Melakukan Inventarisasi Sistem Informasi	47
4.1.18	Use case specification: Mengelola User	48
4.1.19	Use case specification: Mengelola Ref Arsitektur Aplikasi.....	50
4.1.20	Use case specification: Mengelola Ref Bahasa pemograman	52
4.1.21	Use case specification: Mengelola Ref Database server	54
4.1.22	Use case specification: Mengelola Ref Jenis APP	56
4.1.23	Use case specification: Mengelola Ref Model fitur	58
4.1.24	Use case specification: Mengelola Ref Sistem operasi.....	60
4.1.25	Use case specification: Mengelola Ref Status	61
4.1.26	Use case specification: Mengelola Ref Tools pengembangan.....	63
5.	ENTITY RELATIONSHIP DIAGRAM (ERD)	66

Daftar Gambar

Gambar 1. Arsitektur Perangkat Lunak SIKSI.....	11
Gambar 2. Use Case Diagram.....	23
Gambar 3. Entity Relationship Diagram.....	66



1. Pendahuluan

1.1 Tujuan

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) ini merupakan dokumen spesifikasi kebutuhan perangkat lunak SIKSI (Sistem Informasi Kantor Sistem Informasi) untuk mendefinisikan kebutuhan perangkat lunak yang meliputi antarmuka eksternal (antarmuka antara sistem dengan sistem lain perangkat lunak dan pengguna), layanan-layanan yang diberikan dalam sistem, atribut (*feature-feature* tambahan yang dimiliki sistem), serta mendefinisikan fungsi perangkat lunak. SKPL-SIKSI ini juga mendefinisikan batasan perancangan perangkat lunak.

1.2 Lingkup Masalah

Perangkat Lunak SIKSI dikembangkan dengan tujuan untuk :

1. Menangani pengajuan permohonan pembuatan aplikasi dalam lingkup Universitas Atma Jaya Yogyakarta.
2. Menangani pengajuan penambahan fitur aplikasi.
3. Menangani proses persetujuan permohonan pembuatan aplikasi.
4. Menangani proses pengelolaan permohonan pembuatan aplikasi.
5. Menangani proses pengelolaan penambahan fitur aplikasi.
6. Menyediakan pemantauan kinerja staff KSI.
7. Menyediakan laporan untuk proyek manajer.
8. Menyediakan notifikasi penugasan proyek untuk staff IT.
9. Menyediakan laporan CCP untuk staff IT.

Program Studi Teknik Informatika	SKPL – SIKSI	7/ 66
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

10. Menangani proses inventarisasi sistem/aplikasi dan menyediakan laporan deskripsi sistem/aplikasi.

1.3 Definisi, akronim, dan singkatan

Daftar definisi akronim dan singkatan :

Keyword/Phrase	Definisi
SKPL	Merupakan spesifikasi kebutuhan dari perangkat lunak yang akan dikembangkan.
SKPL-SIKSI-XXX	Kode yang merepresentasikan kebutuhan pada SIKSI (Sistem Informasi Kantor Sistem Informasi) dimana XXX merupakan nomor fungsi produk.
SIKSI	Sistem Informasi Kantor Sistem Informasi
KSI	Kantor Sistem Informasi
Internet	Internet merupakan istilah umum yang dipakai untuk menunjuk <i>Network</i> global yang terdiri dari komputer dan layanan servis dengan sekitar 30 sampai 50 juta pemakai komputer dan puluhan layanan informasi termasuk e-mail, FTP, dan <i>World Wide Web</i> .
Server	Komputer yang menyediakan sumber daya bagi klien yang terhubung melalui jaringan.
<i>Client</i>	Komputer yang menghubungi <i>server</i> melalui jaringan

<i>Web based</i>	Menyatakan aplikasi yang dalam penggunaannya pada komputer client dapat diakses mempergunakan web browser tanpa memerlukan proses instalasi terlebih dahulu.
IIS	Internet Information Services HTTP web server yang digunakan dalam sistem operasi server Windows.
<i>Privilege</i>	Menyatakan peran pengguna di dalam sistem yang menyangkut hak akses pengguna tersebut.
<i>CCP</i>	<i>Curriculum Credit Point</i>

1.4 Referensi

Referensi yang digunakan pada perangkat lunak tersebut adalah:

1. Bennet Simon, McRobb Steve, Farmer Ray, *Object-Oriented Sistem Analysis and Design Using UML*, McGraw-Hill Companies, 2002.
2. Jefri Trio Vember, Spesifikasi Kebutuhan Perangkat Lunak (SKPL), Jurusan Teknik Informatika UAJY.

1.5 Deskripsi Umum(Overview)

Secara umum dokumen SKPL ini terbagi atas 2 bagian utama. Bagian utama berisi penjelasan mengenai dokumen SKPL tersebut yang mencakup tujuan pembuatan SKPL, ruang lingkup masalah dalam pengembangan perangkat

lunak tersebut, definisi, referensi dan deskripsi umum tentang dokumen SKPL ini.

Bagian kedua berisi penjelasan umum tentang perangkat lunak SIKSI yang akan dikembangkan, mencakup perspektif produk yang akan dikembangkan, fungsi produk perangkat lunak, karakteristik pengguna, batasan dalam penggunaan perangkat lunak dan asumsi yang dipakai dalam pengembangan perangkat lunak SIKSI tersebut.

2. Deskripsi Kebutuhan

2.1 Perspektif Produk

SIKSI merupakan perangkat lunak yang dikembangkan untuk membantu KSI menangani pengelolaan aplikasi yang dikembangkan oleh KSI. Sistem ini diharapkan dapat melakukan pengelolaan data mulai dari pengajuan permohonan pembuatan aplikasi baru, pengajuan penambahan fitur aplikasi, manajemen proyek yang dikerjakan, penyediaan laporan kinerja staff IT, dan laporan pengembangan aplikasi.

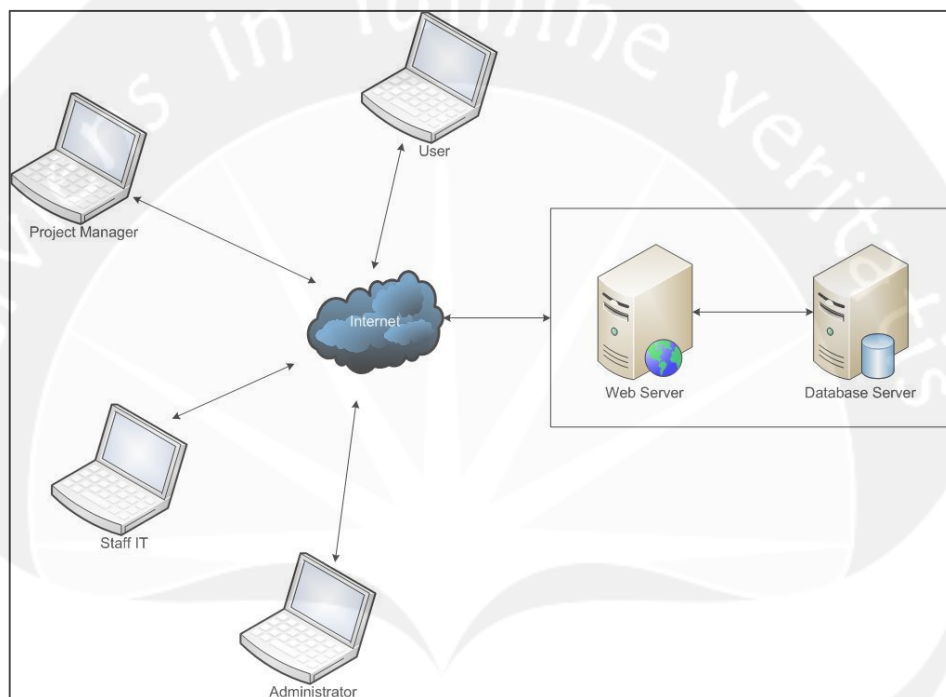
Perangkat lunak ini berjalan pada platform Web ASPX, dikembangkan pada lingkungan sistem operasi windows 7 dengan menggunakan IIS dan dibuat menggunakan bahasa pemrograman C#, XHTML, dan Javascript (*framework* JQuery), dan lingkungan pemrogramannya menggunakan tools Microsoft Visual Studio 2010.

Perangkat lunak ini dapat diakses oleh client pada lingkungan sistem operasi apapun yang mempunyai aplikasi penjelajah situs (*web browser*).

Pengguna akan berinteraksi dengan sistem melalui antarmuka GUI (*Graphical User Interface*). Pada sistem

Program Studi Teknik Informatika	SKPL – SIKSI	10/ 66
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

ini, seperti terlihat pada gambar 1, arsitektur perangkat lunak yang digunakan berupa *client-server*, di mana semua data disimpan di *server*. *Client* dapat mengakses data yang ada di *server* tersebut secara *online* dengan membuka *web site* yang merujuk pada sistem yang telah ditanamkan pada *web server*.



Gambar 1. Arsitektur Perangkat Lunak SIKSI

2.2 Fungsi Produk

Fungsi produk perangkat lunak SIKSI adalah sebagai berikut:

1. Fungsi *Login* (SKPL-SIKSI-001)

Merupakan fungsi yang digunakan oleh pengguna untuk dapat masuk dalam sistem dan mendapatkan hak akses sesuai dengan *privilege* yang dimiliki.

2. Fungsi *Pengajuan Permohonan Pembuatan Aplikasi Baru* (SKPL-SIKSI-002)

Program Studi Teknik Informatika	SKPL – SIKSI	11/ 66
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Merupakan fungsi yang digunakan oleh pengguna untuk mengajukan permohonan pengembangan aplikasi baru.

3. Fungsi *Pengajuan Penambahan Fitur* (**SKPL-SIKSI-003**)

Merupakan fungsi yang digunakan oleh pengguna untuk mengajukan permohonan penambahan proses bisnis dan penambahan fitur aplikasi yang sudah dimiliki.

4. Fungsi *Project Management* (**SKPL-SIKSI-004**)

Merupakan fungsi yang digunakan oleh proyek manajer untuk melakukan persetujuan terhadap pengajuan permohonan pembuatan aplikasi dan pengelolaan proyek pengembangan aplikasi.

Fungsi ini meliputi:

a. Fungsi *Approve Permohonan Pengajuan Pembuatan Aplikasi* (**SKPL-SIKSI-004-01**)

Merupakan fungsi yang digunakan untuk menyetujui permohonan pembuatan aplikasi

b. Fungsi *Tambah Fitur Aplikasi* (**SKPL-SIKSI-004-02**)

Merupakan fungsi yang digunakan untuk menambahkan dan medeskripsikan fitur aplikasi yang sudah disetujui.

5. Fungsi *Pengelolaan fitur project* (**SKPL-SIKSI-005**)

Merupakan fungsi yang digunakan oleh proyek manajer untuk mengolongkan fitur-fitur yang ada kedalam beberapa 3 tipe pengerjaan yaitu *Assign as Form*, *Assign as Report*, dan *Assign as Design*

6. Fungsi *Pengelolaan Form Aplikasi* (**SKPL-SIKSI-006**)

Merupakan fungsi yang digunakan oleh proyek manajer untuk menugaskan *staff IT* untuk mengerjakan fitur-fitur yang telah ditetapkan sebagai *form*.

7. Fungsi *Pengelolaan Report Aplikasi* (**SKPL-SIKSI-007**)

Merupakan fungsi yang digunakan oleh proyek manajer untuk menugaskan *staff IT* untuk mengerjakan fitur-fitur yang telah ditetapkan sebagai *report*.

8. Fungsi *Pengelolaan Design Aplikasi* (**SKPL-SIKSI-008**)

Merupakan fungsi yang digunakan oleh proyek manajer untuk menugaskan *staff IT* untuk mengerjakan fitur-fitur yang telah ditetapkan sebagai *design*.

9. Fungsi *Maintenance Aplikasi* (**SKPL-SIKSI-009**)

Merupakan fungsi yang digunakan oleh proyek manajer untuk melihat permohonan penambahan fitur yang diajukan dan melakukan persetujuan.

10. Fungsi *Monitoring resource* (**SKPL-SIKSI-010**)

Merupakan fungsi yang digunakan oleh proyek manajer untuk memantau *staff IT*.

11. Fungsi *Cetak Laporan Project Manajer* (**SKPL-SIKSI-11**)

Merupakan fungsi yang digunakan oleh proyek manajer untuk mencetak laporan *staff summary* dan laporan detail aplikasi dalam jangka waktu tertentu.

12. Fungsi *Laporan Perkembangan Proyek* (**SKPL-SIKSI-012**)

Merupakan fungsi yang digunakan oleh proyek manajer untuk melihat perkembangan proyek dalam bentuk *tree view*.

13. Fungsi *Penugasan Proyek* (**SKPL-SIKSI-013**)

Merupakan fungsi yang digunakan oleh *staff IT* untuk memeriksa penugasan yang ditujukan padanya dan memutuskan untuk mengerjakannya atau menundanya.

14. Fungsi *Task* (**SKPL-SIKSI-014**)

Merupakan fungsi yang digunakan oleh *staff IT* untuk melihat tugas yang sedang dikerjakan, dan tugas yang sudah selesai dikerjakan serta meng-*update* status tugas menjadi selesai dikerjakan.

15. Fungsi *Cetak Surat CCP* (**SKPL-SIKSI-015**)

Merupakan fungsi yang digunakan oleh *staff IT* untuk mencetak surat CCP.

16. Fungsi *Laporan Sistem Informasi* (**SKPL-SIKSI-016**)

Merupakan fungsi yang digunakan oleh proyek manajer untuk melihat detail aplikasi yang sudah dikembangkan

17. Fungsi *Inventarisasi Sistem Informasi* (**SKPL-SIKSI-017**)

Merupakan fungsi yang digunakan oleh proyek manajer untuk melakukan inventarisasi dari aplikasi yang sudah selesai dikerjakan.

18. Fungsi *Kelola User* (**SKPL-SIKSI-018**)

Merupakan fungsi yang digunakan administrator untuk mengelola user.

Fungsi kelola user mencakup:

a. Fungsi *View* (**SKPL-SIKSI-018-01**)

Fungsi yang digunakan untuk menampilkan data pengguna.

b. Fungsi *Tambah User* (**SKPL-SIKSI-018-02**)

Fungsi yang digunakan untuk menambah pengguna.

c. Fungsi *Reset Password* (**SKPL-SIKSI-018-03**)

Fungsi yang digunakan untuk mengubah password pengguna menjadi password standar.

d. Fungsi *Delete User* (**SKPL-SIKSI-018-03**)

Fungsi yang digunakan untuk menghapus pengguna.

e. Fungsi *Edit privilege* (**SKPL-SIKSI-018-04**)

Fungsi yang digunakan untuk mengubah hak akses pengguna.

19. Fungsi *Kelola Ref Arsitektur Aplikasi* (**SKPL-SIKSI-019**)

Merupakan fungsi yang digunakan administrator untuk mengelola referensi arsitektur aplikasi.

Fungsi kelola ref arsitektur aplikasi mencakup:

a. Fungsi *View* (**SKPL-SIKSI-019-01**)

Fungsi yang digunakan untuk menampilkan data ref arsitektur aplikasi.

b. Fungsi *Tambah Ref Arsitektur Aplikasi* (**SKPL-SIKSI-019-02**)

Program Studi Teknik Informatika	SKPL – SIKSI	15/ 66
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Fungsi yang digunakan untuk menambah ref arsitektur aplikasi.

c. Fungsi *Edit Ref Arsitektur Aplikasi* (**SKPL-SIKSI-019-03**)

Fungsi yang digunakan untuk mengubah data ref arsitektur aplikasi.

20. Fungsi *Kelola Ref Bahasa Pemograman* (**SKPL-SIKSI-020**)

Merupakan fungsi yang digunakan administrator untuk mengelola refrensi bahasa pemograman.

Fungsi kelola ref bahasa pemograman mencakup:

a. Fungsi *View* (**SKPL-SIKSI-020-01**)

Fungsi yang digunakan untuk menampilkan data ref bahasa pemograman.

b. Fungsi *Tambah ref Bahasa Pemograman* (**SKPL-SIKSI-020-02**)

Fungsi yang digunakan untuk menambah ref bahasa pemograman.

c. Fungsi *Edit Ref Bahasa pemograman* (**SKPL-SIKSI-020-03**)

Fungsi yang digunakan untuk mengubah data ref bahasa pemograman.

21. Fungsi *Kelola Ref Database Server* (**SKPL-SIKSI-021**)

Merupakan fungsi yang digunakan administrator untuk mengelola refrensi database server.

Fungsi kelola ref database server mencakup:

a. Fungsi *View* (**SKPL-SIKSI-021-01**)

Fungsi yang digunakan untuk menampilkan data ref database server.

b. Fungsi *Tambah ref Database server* (**SKPL-SIKSI-021-02**)

Fungsi yang digunakan untuk menambah ref database server.

c. Fungsi *Edit Ref Database server* (**SKPL-SIKSI-021-03**)

Fungsi yang digunakan untuk mengubah data ref database server.

22. Fungsi *Kelola Ref Jenis App* (**SKPL-SIKSI-022**)

Merupakan fungsi yang digunakan administrator untuk mengelola referensi jenis aplikasi.

Fungsi kelola ref jenis aplikasir mencakup:

a. Fungsi *View* (**SKPL-SIKSI-022-01**)

Fungsi yang digunakan untuk menampilkan data ref jenis aplikasi.

b. Fungsi *Tambah ref Jenis aplikasi* (**SKPL-SIKSI-022-02**)

Fungsi yang digunakan untuk menambah ref jenis aplikasi.

c. Fungsi *Edit Ref Jenis aplikasi* (**SKPL-SIKSI-022-03**)

Fungsi yang digunakan untuk mengubah data ref jenis aplikasi.

23. Fungsi *Kelola Ref Model Fitur* (**SKPL-SIKSI-023**)

Merupakan fungsi yang digunakan administrator untuk mengelola referensi model fitur.

Fungsi kelola ref model fiturr mencakup:

a. Fungsi *View* (**SKPL-SIKSI-023-01**)

Fungsi yang digunakan untuk menampilkan data ref model fitur.

b. Fungsi *Tambah ref Model fitur* (**SKPL-SIKSI-023-02**)

Fungsi yang digunakan untuk menambah ref model fitur.

c. Fungsi *Edit Ref Model fitur* (**SKPL-SIKSI-023-03**)

Fungsi yang digunakan untuk mengubah data ref model fitur.

24. Fungsi *Kelola Ref Sistem Operasi* (**SKPL-SIKSI-024**)

Merupakan fungsi yang digunakan administrator untuk mengelola referensi sistem operasi.

Fungsi kelola ref sistem operasi mencakup:

a. Fungsi *View* (**SKPL-SIKSI-024-01**)

Fungsi yang digunakan untuk menampilkan data ref sistem operasi.

b. Fungsi *Tambah ref Sistem operasi* (**SKPL-SIKSI-024-02**)

Fungsi yang digunakan untuk menambah ref sistem operasi.

c. Fungsi *Edit Ref Sistem operasi* (**SKPL-SIKSI-024-03**)

Fungsi yang digunakan untuk mengubah data ref sistem operasi.

25. Fungsi *Kelola Ref Status* (**SKPL-SIKSI-025**)

Merupakan fungsi yang digunakan administrator untuk mengelola referensi status.

Fungsi kelola ref status mencakup:

a. Fungsi *View* (**SKPL-SIKSI-025-01**)

Fungsi yang digunakan untuk menampilkan data ref status.

b. Fungsi *Tambah ref Status* (**SKPL-SIKSI-025-02**)

Fungsi yang digunakan untuk menambah ref status.

c. Fungsi *Edit Ref Status* (**SKPL-SIKSI-025-03**)

Fungsi yang digunakan untuk mengubah data ref status.

26. Fungsi *Kelola Ref Tools Pengembangan* (**SKPL-SIKSI-026**)

Merupakan fungsi yang digunakan administrator untuk mengelola refrensi tools pengembangan.

Fungsi kelola ref tools pengembangan mencakup:

a. Fungsi *View* (**SKPL-SIKSI-026-01**)

Fungsi yang digunakan untuk menampilkan data ref tools pengembangan.

b. Fungsi *Tambah ref Tools pengembangan* (**SKPL-SIKSI-026-02**)

Fungsi yang digunakan untuk menambah ref tools pengembangan.

c. Fungsi *Edit Ref Tools pengembangan* (**SKPL-SIKSI-026-03**)

Fungsi yang digunakan untuk mengubah data ref tools pengembangan.

2.3 Karakteristik Pengguna

Karakteristik dari pengguna perangkat lunak SIKSI adalah sebagai berikut :

1. Administrator.

- a. Memahami pengoperasian komputer.
- b. Mengerti tentang internet.
- c. Memahami pengoperasian *web browser*.

Program Studi Teknik Informatika	SKPL – SIKSI	19/ 66
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

2. Staff IT

- a. Memahami pengoperasian komputer.
- b. Mengerti tentang internet.
- c. Memahami pengoperasian *web browser*.

3. Staff

- a. Memahami pengoperasian komputer.
- b. Mengerti tentang internet.
- c. Memahami pengoperasian *web browser*.

4. Proyek Manajer

- a. Memahami pengoperasian komputer.
- b. Mengerti tentang internet.
- c. Memahami pengoperasian *web browser*.

2.4 Batasan-batasan

Batasan-batasan dalam pengembangan perangkat lunak SIKSI tersebut adalah :

1. Kebijakan Umum

Berpedoman pada tujuan dari pengembangan perangkat lunak SIKSI UAJY.

2. Keterbatasan perangkat keras

Dapat diketahui kemudian setelah sistem ini berjalan (sesuai dengan kebutuhan).

2.5 Asumsi dan Ketergantungan

Sistem ini dapat dijalankan pada perangkat komputer (PC, Laptop, dll) yang menggunakan sistem operasi apapun dan mempunyai aplikasi penjelajah situs (*web browser*).

Program Studi Teknik Informatika	SKPL – SIKSI	20/ 66
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

3. Kebutuhan Khusus

3.1 Kebutuhan antarmuka eksternal

Kebutuhan antarmuka eksternal pada perangkat lunak SIKSI meliputi kebutuhan antarmuka pemakai, antarmuka perangkat keras, antarmuka perangkat lunak, antarmuka komunikasi.

3.1.1 Antarmuka pemakai

Pengguna berinteraksi dengan antarmuka yang ditampilkan dalam bentuk page atau form.

3.1.2 Antarmuka perangkat keras

Antarmuka perangkat keras yang digunakan dalam perangkat lunak SIKSI adalah:

1. Perangkat komputer (PC, Laptop, dll).
2. Lan Card.

3.1.3 Antarmuka perangkat lunak

Perangkat lunak yang dibutuhkan untuk mengoperasikan perangkat lunak SIKSI di sisi server adalah sebagai berikut :

1. Nama : .NET Framework 4
Sumber : Microsoft
Sebagai kerangka kerja utama pada sistem operasi Windows.
2. Nama : JQuery API
Sumber : GNU GPL dan MIT
Pustaka javascript.
3. Nama : SQL Server 2008
Sumber : Microsoft
Sebagai database management system (DBMS) yang digunakan untuk penyimpan data di sisi server.
4. Nama : Windows Server 2000

Program Studi Teknik Informatika	SKPL – SIKSI	21/ 66
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Sumber : Microsoft
Sebagai sistem operasi untuk server.

5. Nama : IIS
Sumber : Microsoft
Sebagai web server.

Sedangkan perangkat lunak yang dibutuhkan untuk mengoperasikan perangkat lunak SIKSI di sisi client adalah sebagai berikut :

1. Nama : Microsoft Windows 95/98/NT/2000/XP,
Linux , UNIX , MacOS, dll.

Sebagai sistem operasi untuk pengguna umum.

2. Nama : Microsoft Internet Explorer, Netscape
Navigator, Opera, Safari, Mozilla
Firefox, dll.

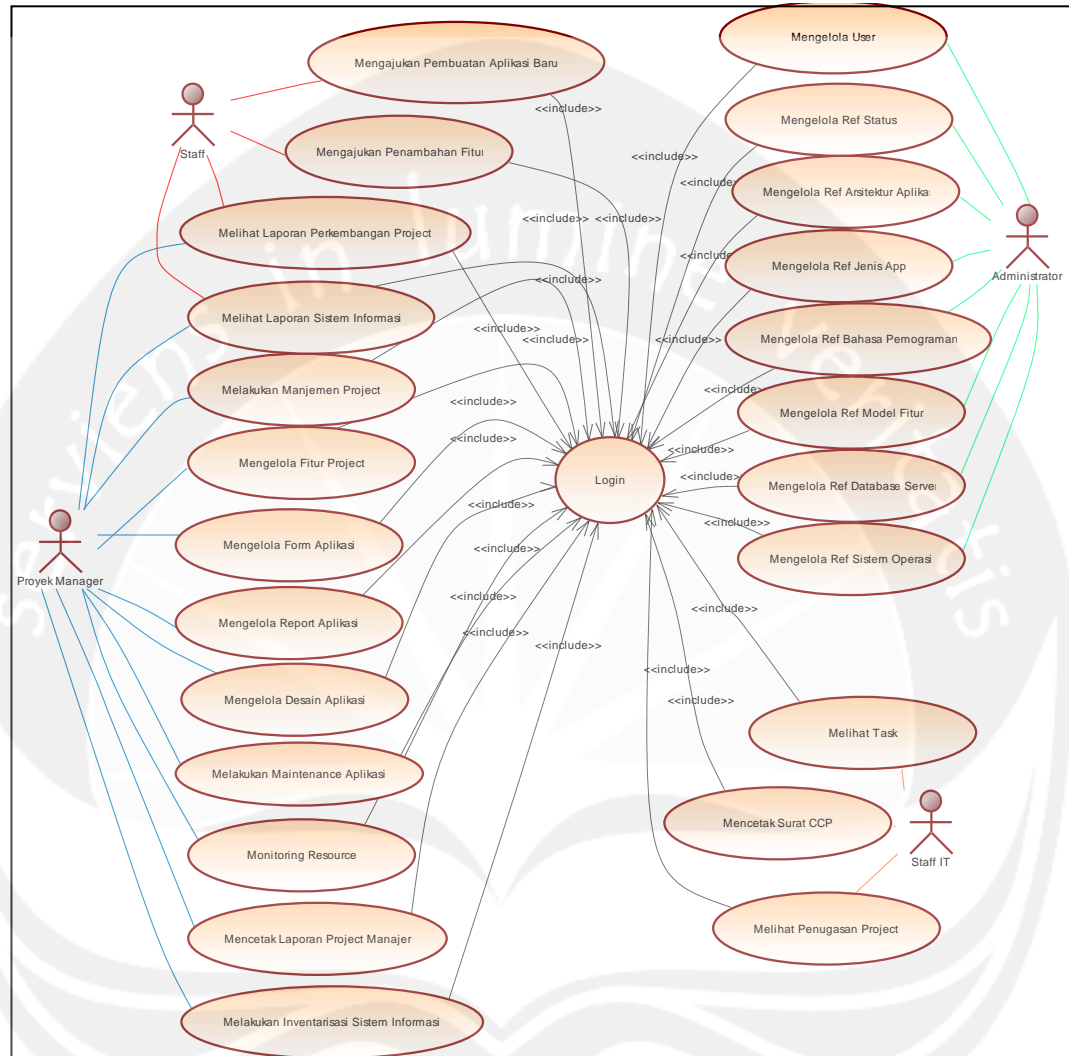
Sebagai penjelajah situs (web browser) berbasis grafis atau teks.

3.1.4 Antarmuka Komunikasi

Antarmuka komunikasi perangkat lunak SIKSI UAJY terhubung secara client-server dalam lingkup jaringan Internet atau intranet berbasis protokol Transmission Control Protocol/Internet Protocol (TCP/IP).

3.2 Kebutuhan Fungsionalitas Perangkat Lunak

3.2.1 Use Case Diagram



Gambar 2. Use Case Diagram

4. Spesifikasi rinci kebutuhan

4.1 Spesifikasi Kebutuhan Fungsionalitas

4.1.1 Use case Specification: Login

1. Brief Description

Use case ini digunakan oleh aktor untuk memperoleh akses ke sistem.

2. Primary Actor

1. Administrator
2. *Staff*
3. *Staff IT*
4. *Project Manajer*

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan login.
2. Sistem menampilkan antarmuka untuk login.
3. Aktor memasukkan id login dan password.
4. Sistem memeriksa id login dan password yang diinputkan aktor.
- E-1 Username dan password login user salah.
5. Sistem memberikan akses ke aktor.
6. Use Case ini selesai.

5. Alternative Flow

none

6. Error Flow

E-1 Password dan id login user tidak sesuai

1. Sistem menampilkan peringatan bahwa password dan id login user tidak sesuai.
- Kembali ke Basic Flow langkah ke-3.

7. PreConditions

1. Sistem berjalan dengan baik dan lancar.
2. Aktor memiliki koneksi internet sehingga dapat terhubung dengan sistem.
3. Sistem berjalan dengan baik dan lancar.
4. Aktor telah terdaftar pada sistem.
5. Aktor telah membuka halaman antarmuka login.

8. PostConditions

1. Aktor dapat memasuki sistem dan dapat menggunakan fungsi-fungsi pada sistem.

4.1.2 Use case Specification: Mengajukan Permohonan Pembuatan Aplikasi

1. Brief Description

Use case ini digunakan oleh aktor untuk mengajukan permohonan pembuatan aplikasi.

2. Primary Actor

1. *Staff*

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk mengajukan permohonan pembuatan aplikasi.
2. Sistem menampilkan antarmuka untuk melakukan pengajuan permohonan pembuatan aplikasi baru
3. Aktor memasukkan data pengajuan permohonan.
4. Sistem memeriksa data pengajuan permohonan yang diinputkan aktor valid atau tidak.
E-1 Data pengajuan tidak valid.

5. Sistem menampilkan antarmuka yang menunjukkan bahwa pengajuan permohonan telah berhasil.

6. Use Case ini selesai.

5. Alternative Flow

none

6. Error Flow

E-1 Data pengajuan tidak valid.

1. Sistem menampilkan peringatan bahwa data pengajuan tidak valid.

2. Kembali ke Basic Flow langkah ke-2.

7. PreConditions

1. Use Case Login telah dilakukan.

2. Aktor telah memasuki sistem.

8. PostConditions

1. Data pengajuan permohonan pembuatan aplikasi baru telah ter-update.

4.1.3 Use case Specification: Mengajukan Penambahan Fitur

1. Brief Description

Use case ini digunakan oleh aktor untuk mengajukan penambahan fitur dan proses bisnis dari aplikasi yang sudah dimiliki.

2. Primary Actor

1. Staff

3. Supporting Actor

none

4. Basic Flow

1. Use case ini dimulai ketika aktor memilih untuk melakukan pengajuan penambahan fitur.

2. Sistem menampilkan antarmuka aplikasi yang dimiliki.
3. Aktor memilih aplikasi yang akan ditambah fiturnya.
4. Sistem menampilkan antarmuka detail aplikasi yang dipilih.
5. Aktor memilih proses bisnis yang akan ditambah fiturnya.
A-1 Aktor memilih menambah proses bisnis.
6. Sistem menampilkan antarmuka untuk melakukan pengajuan penambahan fitur.
7. Aktor memasukkan data pengajuan penambahan fitur.
8. Sistem memeriksa data pengajuan penambahan fitur yang diinputkan aktor valid atau tidak.
E-1 Data pengajuan penambahan fitur tidak valid.
9. Sistem menampilkan antarmuka yang menunjukkan bahwa pengajuan penambahan fitur telah berhasil.
10. Use case ini telah selesai.

5. Alternative Flow

- A-1 Aktor memilih untuk melakukan penambahan proses bisnis
1. Sistem menampilkan antarmuka untuk melakukan penambahan proses bisnis.
 2. Aktor memasukkan data proses bisnis.
 3. Sistem memeriksa data proses bisnis yang dimasukkan aktor valid atau tidak.
- E-2 Data proses bisnis tidak valid.
1. Kembali ke Basic Flow langkah ke-4.

6. Error Flow

- E-1 Data penambahan fitur tidak valid

Program Studi Teknik Informatika	SKPL – SIKSI	27/ 66
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

1. Sistem menampilkan peringatan bahwa data penambahan fitur tidak valid.

2. Kembali ke Basic Flow langkah ke-6.

E-2 Data penambahan proses bisnis tidak valid

1. Sistem menampilkan peringatan bahwa data penambahan proses bisnis tidak valid.

2. Kembali ke Alternative Flow langkah ke-1.

7. PreConditions

1. Use Case Login telah dilakukan.

2. Aktor telah memasuki sistem.

8. PostConditions

1. Data penambahan fitur aplikasi didatabase telah ter-update.

4.1.4 Use case Specification: Melakukan Manajemen Project

1. Brief Description

Use case ini digunakan oleh aktor untuk melakukan persetujuan terhadap pengajuan permohonan pembuatan aplikasi dan pengelolaan *project* pengembangan aplikasi.

2. Primary Actor

1. *Project* Manajer

3. Supporting Actor

none

4. Basic Flow

1. Use case ini dimulai ketika aktor memilih untuk melakukan manajemen *project*.

2. Sistem menampilkan antarmuka permohonan pengembangan aplikasi yang diajukan dan pilihan tambah fitur aplikasi.

Program Studi Teknik Informatika	SKPL – SIKSI	28/ 66
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

3. Aktor melakukan persetujuan permohonan pengembangan aplikasi.

A-1 Aktor memilih pilihan tambah fitur aplikasi.

4. Use case ini telah selesai.

5. Alternative Flow

A-1 Aktor memilih pilihan tambah fitur aplikasi.

1. Sistem menampilkan antarmuka aplikasi yang ada.
2. Aktor memilih salah satu aplikasi

A-2 Aktor memasukkan nama aplikasi ke dalam kolom kata kunci pencarian aplikasi.

1. Sistem menampilkan daftar proses bisnis dari aplikasi yang dipilih.
2. Aktor memilih proses bisnis yang akan ditambah fiturnya.
3. Sistem menampilkan antarmuka berupa data fitur yang merupakan bagian dari proses bisnis yang dipilih.
4. Aktor memilih menambahkan pilihan tambah fitur.
5. Sistem menampilkan antarmuka untuk melakukan penambahan fitur.
6. Aktor memasukkan data fitur.
7. Sistem memeriksa data yang diinputkan valid atau tidak.

E-1 Data fitur tidak valid.

1. Sistem menampilkan antarmuka yang menunjukkan bahwa penambahan fitur telah berhasil.
2. Kembali ke Alternative Flow langkah ke-5

A-2 Aktor memasukkan nama aplikasi ke dalam kolom kata kunci pencarian aplikasi.

1. Sistem menampilkan hasil pencarian yang ada.

E-2 Hasil pencarian yang sesuai tidak ditemukan.

1. Kembali ke Alternative Flow langkah ke-2.

6. Error Flow

E-1 Data fitur tidak valid

1. Sistem menampilkan peringatan bahwa data fitur tidak valid.

2. Kembali ke Alternative Flow langkah ke-7.

E-2 Hasil pencarian yang sesuai tidak ditemukan.

1. Kembali ke Alternative Flow langkah ke-2.

7. PreConditions

1. Use Case Login telah dilakukan.

2. Aktor telah memasuki sistem.

8. PostConditions

1. Data aplikasi, Fitur dan pengajuan permohonan di database telah ter-update.

4.1.5 Use case Specification: Mengelola Fitur Project

1. Brief Description

Use case ini digunakan oleh aktor untuk mengolompokkan fitur-fitur yang ada ke dalam 3 tipe pengerjaan.

2. Primary Actor

1. *Project Manajer*

3. Supporting Actor

none

4. Basic Flow

1. Use case ini dimulai ketika aktor memilih untuk melakukan pengelolaan fitur *project*.

Program Studi Teknik Informatika	SKPL – SIKSI	30/ 66
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

2. Sistem menampilkan kolom pencarian aplikasi berdasarkan kata kunci.

3. Aktor memasukkan kata kunci.

E-1 Tidak ada hasil pencarian yang sesuai dengan kata kunci yang dimasukkan.

4. Sistem menampilkan fitur aplikasi yang ada sesuai dengan kata kunci yang dimasukkan.

5. Aktor memilih menggolongkan fitur sebagai sebuah form.

A-1 Aktor memilih menggolongkan fitur sebagai sebuah report.

A-2 Aktor memilih menggolongkan fitur sebagai sebuah desain.

6. Sistem menampilkan antarmuka untuk menggolongkan fitur sebagai sebuah form.

7. Aktor memasukkan data form.

8. Sistem memeriksa data form yang dimasukkan aktor valid atau tidak.

E-2 Data form tidak valid.

9. Sistem menampilkan antarmuka yang menunjukkan bahwa fitur berhasil digolongkan sebagai form.

10. Use case ini selesai.

5. Alternative Flow

A-1 Aktor memilih menggolongkan fitur sebagai sebuah report.

1. Sistem menampilkan antarmuka untuk menggolongkan fitur sebagai sebuah report.

2. Aktor memasukkan data report.

3. Sistem memeriksa data report yang dimasukkan aktor valid atau tidak.

E-3 Data report tidak valid.

Program Studi Teknik Informatika	SKPL – SIKSI	31/ 66
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

4. Sistem menampilkan antarmuka yang menunjukkan bahwa fitur berhasil digolongkan sebagai report.

A-2 Aktor memilih menggolongkan fitur sebagai sebuah desain.

1. Sistem menampilkan antarmuka untuk menggolongkan fitur sebagai sebuah desain.
2. Aktor memasukkan data desain.
3. Sistem memeriksa data desain yang dimasukkan aktor valid atau tidak.

E-4 Data desain tidak valid.

4. Sistem menampilkan antarmuka yang menunjukkan bahwa fitur berhasil dikategorikan sebagai desain.

6. Error Flow

E-1 Tidak ada hasil pencarian yang sesuai dengan kata kunci yang dimasukkan.

1. Sistem kembali ke Basic Flow langkah ke-2.

E-2 Data form tidak valid.

1. Sistem menampilkan peringatan bahwa data form tidak valid.
2. Kembali ke Basic Flow langkah ke-6.

E-3 Data report tidak valid.

1. Sistem menampilkan peringatan bahwa data report tidak valid.
2. Kembali ke Alternative Flow 1 langkah ke-1.

E-4 Data desain tidak valid.

1. Sistem menampilkan peringatan bahwa data desain tidak valid
2. Kembali ke Alternative Flow 2 langkah ke-1.

7. PreConditions

1. Use Case Login telah dilakukan.
2. Aktor telah memasuki sistem.

8. PostConditions

1. Data form, report, desain, dan fitur di database telah ter-update.

4.1.6 Use case Specification: Mengelola Form Aplikasi

1. Brief Description

Use case ini digunakan oleh aktor untuk melakukan penugasan kepada staff IT atau outsource.

2. Primary Actor

1. *Project Manajer*

3. Supporting Actor

none

4. Basic Flow

1. Use case ini dimulai ketika aktor memilih melakukan pengelolaan form aplikasi.
2. Sistem menampilkan antarmuka data aplikasi.
3. Aktor memilih aplikasi.
4. Sistem menampilkan antarmuka fitur yang terdapat pada aplikasi yang dipilih.
5. Aktor memilih fitur.
E-1 Fitur yang dipilih tidak digolongkan sebagai form.
6. Aktor memilih penugasan kepada staff.
A-1 Aktor memilih penugasan kepada outsource.
7. Sistem menampilkan antarmuka penugasan kepada staff.
8. Aktor memasukkan data penugasan kepada staff.

9. Sistem memeriksa data yang dimasukkan aktor valid atau tidak.

E-2 Data yang penugasan form yang dimasukkan tidak valid.

10. Sistem menampilkan antarmuka yang menunjukkan form berhasil ditugaskan.

11. Use case ini selesai.

5. Alternative Flow

A-1 Aktor memilih penugasan kepada outsource.

1. Sistem menampilkan antarmuka penugasan kepada outsource.

2. Aktor memasukkan data penugasan kepada outsource.

3. Sistem memeriksa data yang dimasukkan aktor valid atau tidak.

E-3 Data yang penugasan form yang dimasukkan tidak valid.

4. Sistem menampilkan antarmuka yang menunjukkan form berhasil ditugaskan.

5. Berlanjut ke Basic Flow langkah ke-11.

6. Error Flow

E-1 Fitur yang dipilih tidak digolongkan sebagai form.

1. Sistem menampilkan data kosong.

2. Kembali ke Basic Flow langkah ke-4.

E-2 Data yang penugasan form kepada staff yang dimasukkan tidak valid.

1. Sistem menampilkan peringatan bahwa data penugasan form tidak valid.

2. Kembali ke Basic Flow langkah ke-7.

E-3 Data yang penugasan form kepada outsource yang dimasukkan tidak valid.

Program Studi Teknik Informatika	SKPL – SIKSI	34/ 66
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

1. Sistem menampilkan peringatan bahwa data penugasan form tidak valid.

2. Kembali ke Basic Alternative Flow langkah ke-1

7. PreConditions

1. Use Case Login telah dilakukan.

2. Aktor telah memasuki sistem.

8. PostConditions

1. Data form didatabase telah terupdate.

4.1.7 Use case Specification: Mengelola Report Aplikasi

1. Brief Description

Use case ini digunakan oleh aktor untuk melakukan penugasan terhadap staff IT atau outsource.

2. Primary Actor

1. *Project* Manajer

3. Supporting Actor

none

4. Basic Flow

1. Use case ini dimulai ketika aktor memilih melakukan pengelolaan report aplikasi.

2. Sistem menampilkan antarmuka data aplikasi.

3. Aktor memilih aplikasi.

4. Sistem menampilkan antarmuka fitur yang terdapat pada aplikasi yang dipilih.

5. Aktor memilih fitur.

E-1 Fitur yang dipilih tidak digolongkan sebagai report.

6. Aktor memilih penugasan kepada staff.

A-1 Aktor memilih penugasan kepada outsource.

7. Sistem menampilkan antarmuka penugasan kepada staff.
8. Aktor memasukkan data penugasan kepada staff.
9. Sistem memeriksa data yang dimasukkan aktor valid atau tidak.
E-2 Data yang penugasan report yang dimasukkan tidak valid.
10. Sistem menampilkan antarmuka yang menunjukkan report berhasil ditugaskan.
11. Use case ini selesai.

5. Alternative Flow

- A-1 Memilih penugasan kepada outsource.
1. Sistem menampilkan antarmuka penugasan kepada outsource.
 2. Aktor memasukkan data penugasan kepada outsource.
 3. Sistem memeriksa data yang dimasukkan aktor valid atau tidak.
E-3 Data yang penugasan report yang dimasukkan tidak valid.
 4. Sistem menampilkan antarmuka yang menunjukkan report berhasil ditugaskan.
 5. Berlanjut ke Basic Flow langkah ke-11.

6. Error Flow

- E-1 Fitur yang dipilih tidak dikategorikan sebagai report.
1. Sistem menampilkan data kosong.
 2. Kembali ke Basic Flow langkah ke-4.
- E-2 Data yang penugasan report yang dimasukkan tidak valid.

1. Sistem menampilkan peringatan bahwa data penugasan report tidak valid.

2. Kembali ke Basic Flow langkah ke-7.

E-3 Data yang penugasan report yang dimasukkan tidak valid.

1. Sistem menampilkan peringatan bahwa data penugasan report tidak valid.

2. Kembali ke Basic Alternative Flow langkah ke-1

7. PreConditions

1. Use Case Login telah dilakukan.

2. Aktor telah memasuki sistem.

8. PostConditions

1. Data report didatabase telah ter-update.

4.1.8 Use case Specification: Mengelola Desain Aplikasi

1. Brief Description

Use case ini digunakan oleh aktor untuk mengelola desain dan melakukan penugasan terhadap staff IT atau outsource.

2. Primary Actor

1. *Project Manajer*

3. Supporting Actor

none

4. Basic Flow

1. Use case ini dimulai ketika aktor memilih melakukan pengelolaan desain aplikasi.

2. Sistem menampilkan antarmuka data aplikasi.

3. Aktor memilih aplikasi.

4. Sistem menampilkan antarmuka fitur yang terdapat pada aplikasi yang dipilih.
5. Aktor memilih fitur.
E-1 Fitur yang dipilih tidak dikategorikan sebagai desain.
6. Aktor memilih penugasan kepada staff.
A-1 Aktor memilih penugasan kepada outsource.
7. Sistem menampilkan antarmuka penugasan kepada staff.
8. Aktor memasukkan data penugasan kepada staff.
9. Sistem memeriksa data yang dimasukkan aktor valid atau tidak.
E-2 Data yang penugasan desain yang dimasukkan tidak valid.
10. Sistem menampilkan antarmuka yang menunjukkan desain berhasil ditugaskan.
11. Use case ini selesai.

5. Alternative Flow

- A-1 memilih penugasan kepada outsource.
1. Sistem menampilkan antarmuka penugasan kepada outsource.
 2. Aktor memasukkan data penugasan kepada outsource.
 3. Sistem memeriksa data yang dimasukkan aktor valid atau tidak.
E-3 Data yang penugasan desain yang dimasukkan tidak valid.
 4. Sistem menampilkan antarmuka yang menunjukkan desain berhasil ditugaskan.
 5. Berlanjut ke Basic Flow langkah ke-11.

6. Error Flow

E-1 Fitur yang dipilih tidak dikategorikan sebagai desain.

1. Sistem menampilkan data kosong.
2. Kembali ke Basic Flow langkah ke-4.

E-2 Data yang penugasan desain yang dimasukkan tidak valid.

1. Sistem menampilkan peringatan bahwa data penugasan desain tidak valid.
2. Kembali ke Basic Flow langkah ke-7.

E-3 Data yang penugasan desain yang dimasukkan tidak valid.

1. Sistem menampilkan peringatan bahwa data penugasan desain tidak valid.
2. Kembali ke Basic Alternative Flow langkah ke-1

7. PreConditions

1. Use Case Login telah dilakukan.
2. Aktor telah memasuki sistem.

8. PostConditions

1. Data desain didatabase telah terupdate.

4.1.9 Use case Specification: Melakukan Maintenance Aplikasi

1. Brief Description

Use case ini digunakan oleh aktor untuk menyetujui pengajuan penambahan fitur aplikasi.

2. Primary Actor

1. *Project* Manajer

3. Supporting Actor

none

Program Studi Teknik Informatika	SKPL – SIKSI	39/ 66
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

4. Basic Flow

1. Use case ini dimulai ketika aktor memilih untuk melakukan maintenance aplikasi.
2. Sistem menampilkan data pengajuan penambahan fitur.
3. Aktor memilih data pengajuan penambahan fitur.
4. Sistem menampilkan antarmuka detail pengajuan penambahan fitur dan tombol persetujuan.
5. Aktor memilih tombol persetujuan.
6. Sistem menampilkan antarmuka yang menunjukkan pengajuan penambahan fitur telah disetujui.
7. Use case ini selesai.

5. Alternative Flow

none

6. Error Flow

none

7. PreConditions

1. Use Case Login telah dilakukan.
2. Aktor telah memasuki sistem.

8. PostConditions

1. Data pengajuan penambahan fitur, fitur di database telah ter-update.

4.1.10 Use case Specification: Monitoring Resource

1. Brief Description

Use case ini digunakan oleh aktor untuk untuk memantau staff IT.

2. Primary Actor

1. *Project* Manajer

3. Supporting Actor

none

Program Studi Teknik Informatika	SKPL – SIKSI	40/ 66
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

4. Basic Flow

1. Use case ini dimulai ketika aktor memilih untuk melakukan pemantauan resource.
2. Sistem menampilkan data pekerjaan yang sedang dikerjakan dan data staff yang tersedia.
3. Use case ini selesai.

5. Alternative Flow

none

6. Error Flow

none

7. PreConditions

1. Use Case Login telah dilakukan.
2. Aktor telah memasuki sistem.

8. PostConditions

1. Data resource telah tertampil.

4.1.11 Use case Specification: Mencetak Laporan Project Manager

1. Brief Description

Use case ini digunakan oleh aktor untuk mencetak laporan staff summary dan laporan detail aplikasi dalam jangka waktu tertentu.

2. Primary Actor

1. *Project* Manajer

3. Supporting Actor

none

4. Basic Flow

1. Use case ini dimulai ketika aktor memilih untuk mencetak laporan *project* manager.
2. Sistem menampilkan antarmuka pilihan untuk mencetak laporan *project* manajer.

3. Aktor memasukkan jangka waktu tanggal dan memilih cetak staff summary.

A-1 Aktor memasukkan jangka waktu tanggal dan memilih cetak laporan detail aplikasi.

4. Sistem menampilkan laporan.

5. Use case ini selesai.

5. Alternative Flow

A-1 Aktor memasukkan jangka waktu tanggal dan memilih cetak laporan detail aplikasi.

1. Sistem menampilkan laporan

2. Berlanjut ke Basic Flow langkah ke-5

6. Error Flow

none

7. PreConditions

1. Use Case Login telah dilakukan.

2. Aktor telah memasuki sistem.

8. PostConditions

1. Laporan *Project Manager* telah tertampil.

4.1.12 Use case Specification: Melihat Laporan Perkembangan Project

1. Brief Description

Use case ini digunakan oleh aktor untuk melihat perkembangan *project*.

2. Primary Actor

1. *Project Manajer*

3. Supporting Actor

none

4. Basic Flow

1. Use case ini dimulai ketika aktor memilih untuk melihat Laporan perkembangan *project*.

2. Sistem menampilkan data laporan perkembangan *project*.
3. Aktor memilih salah satu aplikasi.
4. Sistem menampilkan fitur yang dimiliki aplikasi.
5. Aktor memilih salah satu fitur.
6. Sistem menampilkan tipe pengerjaan.
7. Aktor memilih tipe pengerjaan.
8. Sistem menampilkan detail tipe pengerjaan.
9. Use case ini selesai.

5. Alternative Flow

none

6. Error Flow

none

7. PreConditions

1. Use Case Login telah dilakukan.
2. Aktor telah memasuki sistem.

8. PostConditions

1. Data perkembangan project telah tertampil dalam bentuk *tree view*.

4.1.13 Use case Specification: Menugaskan Project

1. Brief Description

Use case ini digunakan oleh aktor untuk memeriksa penugasan yang ditunjukkan.

2. Primary Actor

1. Staff IT

3. Supporting Actor

none

4. Basic Flow

1. Use case ini dimulai ketika aktor memilih untuk melihat penugasan *project*.

Program Studi Teknik Informatika	SKPL – SIKSI	43/ 66
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

2. Sistem menampilkan data penugasan *project* yang ditunjukkan pada aktor.
3. Aktor memilih salah satu penugasan.
4. Sistem menampilkan detail penugasan.
5. Aktor memilih untuk mulai mengerjakan tugas yang diberikan.

5. Alternative Flow

none

6. Error Flow

none

7. PreConditions

1. Use Case Login telah dilakukan.
2. Aktor telah memasuki sistem.

8. PostConditions

1. Data form, report, desain di database telah *ter-update*.

4.1.14 Use case specification: Melihat Task

1. Brief Description

Use case ini digunakan oleh aktor untuk melihat tugas yang sedang dikerjakan, dan tugas yang sudah selesai dikerjakan serta memperbarui status tugas menjadi selesai dikerjakan.

2. Primary Actor

1. Staff IT

3. Supporting Actor

none

4. Basic Flow

1. Use case ini dimulai ketika aktor memilih untuk melihat tugas-tugas yang sedang dikerjakan, dan tugas yang sudah selesai dikerjakan.

Program Studi Teknik Informatika	SKPL – SIKSI	44/ 66
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

2. Sistem menampilkan dua pilihan on progress task dan complete task.
3. Aktor memilih on progress task.
A-1 Aktor memilih complete task.
4. Sistem menampilkan tugas yang sedang dikerjakan oleh aktor.
5. Aktor memilih salah satu tugas.
6. Sistem menampilkan detail tugas yang sedang dikerjakan.
7. Aktor memilih untuk menyelesaikan tugas.
8. Use case ini selesai

5. Alternative Flow

A-1 Aktor memilih complete task

1. Sistem menampilkan tugas yang sudah selesai dikerjakan.
2. Aktor memilih salah satu tugas.
3. Sistem menampilkan detail tugas yang dipilih.
4. Berlanjut ke Basic flow langkah ke-8

6. Error Flow

none

7. PreConditions

1. Use Case Login telah dilakukan.
2. Aktor telah memasuki sistem

8. PostConditions

1. Data form, report, desain di database telah ter-update.

4.1.15 Use case specification: Mencetak Surat CCP

1. Brief Description

Use case ini digunakan oleh aktor untuk mencetak surat CCP.

Program Studi Teknik Informatika	SKPL – SIKSI	45/ 66
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

2. Primary Actor

1. Staff IT

3. Supporting Actor

none

4. Basic Flow

1. Use case ini dimulai ketika aktor memilih untuk mencetak surat CCP.
2. Sistem menampilkan antarmuka untuk memasukkan kriteria surat.
3. Aktor memasukkan kriteria surat dan meminta sistem untuk menampilkan surat CCP.
4. Sistem menampilkan surat CCP.
5. Use case selesai.

5. Alternative Flow

none

6. Error Flow

none

7. PreConditions

1. Use Case Login telah dilakukan.
2. Aktor telah memasuki sistem.

8. PostConditions

1. Surat CCP telah tertampil.

4.1.16 Use case specification: Melihat Laporan Sistem Informasi

1. Brief Description

Use case ini digunakan oleh aktor untuk melihat detail aplikasi yang sudah dikembangkan.

2. Primary Actor

1. Project Manajer

3. Supporting Actor

none

4. Basic Flow

1. Use case ini dimulai ketika aktor memilih untuk melihat laporan sistem informasi.
2. Sistem menampilkan data aplikasi.
3. Aktor memilih aplikasi.
4. Sistem menampilkan detail aplikasi.
5. Use case ini selesai.

5. Alternative Flow

none

6. Error Flow

none

7. PreConditions

1. Use Case Login telah dilakukan.
2. Aktor telah memasuki sistem.

8. PostConditions

none

4.1.17 Use case specification: Melakukan Inventarisasi Sistem Informasi

1. Brief Description

Use case ini digunakan oleh aktor untuk melakukan inventarisasi dari aplikasi yang sudah selesai dikerjakan.

2. Primary Actor

1. Proyek Manajer

3. Supporting Actor

none

4. Basic Flow

1. Use case ini dimulai ketika aktor memilih untuk melakukan inventarisasi sistem informasi.
2. Sistem menampilkan data aplikasi yang sedang dikerjakan.
3. Aktor memilih deploy aplikasi.
A-1 Aktor memilih detail aplikasi.
4. Use case ini selesai.

5. Alternative Flow

- A-1 Aktor memilih detail aplikasi.
1. Sistem menampilkan detail aplikasi.
 2. Berlanjut ke Basic flow langkah ke-4.

6. Error Flow

none

7. Preconditions

1. Use Case Login telah dilakukan.
2. Aktor telah memasuki sistem.

8. PostConditions

1. Data aplikasi di database telah ter-update.

4.1.18 Use case specification: Mengelola User

1. Brief Description

Use case ini digunakan oleh aktor untuk mengelola user. Aktor dapat melakukan *view*, *reset password*, dan *delete user*, *edit privilege*.

2. Primary Actor

1. Administrator

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan user.
2. Sistem menampilkan table user.
3. Aktor memilih menambah user.
A-1 Aktor memilih user yang ada pada table user.
4. Sistem menampilkan antarmuka untuk memasukkan data user.
5. Aktor memasukkan data user.
6. Sistem memeriksa data user valid atau tidak.
E-1 Data user tidak valid.
7. Sistem menampilkan antarmuka menunjukkan data user berhasil dimasukkan.
8. Use Case selesai.

5. Alternative Flow

- A-1 Aktor memilih user yang ada pada table user
1. Sistem menampilkan pilihan untuk melakukan reset password, delete, dan edit privilege.
 2. Aktor memilih reset password.
A-2 Aktor memilih delete user.
A-3 Aktor memilih edit privilege
 3. Sistem menampilkan antarmuka yang menunjukkan password berhasil direset.
 4. Berlanjut ke Basic flow langkah ke-8.
- A-2 Aktor memilih delete user.
1. Sistem mengupdate flag user yang dipilih dari database dan menampilkan antarmuka yang menunjukkan user berhasil dihapus.
 2. Berlanjut ke Basic flow langkah ke-8.
- A-3 Aktor memilih edit privilege

1. Aktor memilih privilege yang akan diberikan kepada user.
2. Sistem memeriksa privilege yang dipilih.
- E-2 Tidak ada privilege yang dipilih.
3. Sistem mengupdate data user yang dipilih dan menampilkan antarmuka yang menunjukkan user berhasil diupdate.
4. Berlanjut ke Basic flow langkah ke-8.

6. Error Flow

E-1 Data user tidak valid.

1. Sistem menampilkan pesan peringatan bahwa data user tidak valid.
2. Kembali ke Basic flow langkah ke-4.

E-2 Tidak ada privilege yang dipilih.

1. Sistem menampilkan pesan peringatan bahwa tidak ada privilege yang dipilih.
2. Kembali ke Alternative flow 3 langkah ke-1.

7. PreConditions

1. Use Case Login telah dilakukan.
2. Aktor telah memasuki sistem

8. PostConditions

1. Data user di database telah ter-update.

4.1.19 Use case specification: Mengelola Ref Arsitektur Aplikasi

1. Brief Description

Use case ini digunakan oleh aktor untuk mengelola ref arsitektur aplikasi. Aktor dapat melakukan *view*, *tambah*, dan *edit ref arsitektur aplikasi*.

2. Primary Actor

1. Administrator

Program Studi Teknik Informatika	SKPL – SIKSI	50/ 66
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan *ref arsitektur aplikasi*.
2. Sistem menampilkan list *arsitektur aplikasi*.
3. Aktor memilih menambah arsitektur aplikasi.
A-1 Aktor memilih arsitektur aplikasi yang ada pada list arsitektur aplikasi.
4. Sistem menampilkan antarmuka untuk memasukkan data arsitektur aplikasi.
5. Aktor memasukkan data arsitektur aplikasi.
6. Sistem memeriksa data user valid atau tidak.
E-1 Data arsitektur aplikasi tidak valid.
7. Sistem menampilkan antarmuka menunjukkan data arsitektur aplikasi berhasil dimasukkan.
8. Use Case selesai.

5. Alternative Flow

- A-1 Aktor memilih arsitektur aplikasi yang ada pada list arsitektur aplikasi
1. Sistem menampilkan pilihan untuk melakukan edit.
 2. Aktor memasukkan data arsitektur aplikasi baru yang akan diupdate.
 3. Sistem memeriksa apakah data arsitektur aplikasi valid atau tidak.
E-2 data arsitektur aplikasi tidak valid.
 4. Sistem mengupdate data arsitektur aplikasi yang dipilih dari database dan menampilkan antarmuka yang menunjukkan arsitektur aplikasi berhasil dihapus.

Program Studi Teknik Informatika	SKPL – SIKSI	51/ 66
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

5. Berlanjut ke Basic flow langkah ke-8.

6. Error Flow

E-1 Data arsitektur aplikasi tidak valid.

1. Sistem menampilkan pesan peringatan bahwa data arsitektur aplikasi tidak valid.
2. Kembali ke Basic flow langkah ke-4.

E-2 data arsitektur aplikasi tidak valid

1. Sistem menampilkan pesan peringatan bahwa data arsitektur aplikasi tidak valid.
2. Kembali ke Alternative flow langkah ke-1.

7. PreConditions

1. Use Case Login telah dilakukan.
2. Aktor telah memasuki sistem

8. PostConditions

1. Data ref arsitektur aplikasi di database telah ter-update.

4.1.20 Use case specification: Mengelola Ref Bahasa pemograman

1. Brief Description

Use case ini digunakan oleh aktor untuk mengelola ref bahasa pemograman. Aktor dapat melakukan *view*, *tambah*, dan *edit ref bahasa pemograman*.

2. Primary Actor

1. Administrator

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan *ref bahasa pemograman*.
2. Sistem menampilkan list *bahasa pemograman*.

Program Studi Teknik Informatika	SKPL – SIKSI	52/ 66
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

3. Aktor memilih menambah bahasa pemograman.
A-1 Aktor memilih bahasa pemograman yang ada pada list bahasa pemograman.
4. Sistem menampilkan antarmuka untuk memasukkan data bahasa pemograman.
5. Aktor memasukkan data bahasa pemograman.
6. Sistem memeriksa data user valid atau tidak.
E-1 Data bahasa pemograman tidak valid.
7. Sistem menampilkan antarmuka menunjukkan data bahasa pemograman berhasil dimasukkan.
8. Use Case selesai.

5. Alternative Flow

- A-1 Aktor memilih bahasa pemograman yang ada pada list bahasa pemograman
1. Sistem menampilkan pilihan untuk melakukan edit.
 2. Aktor memasukkan data bahasa pemograman baru yang akan diupdate.
 3. Sistem memeriksa apakah data bahasa pemograman valid atau tidak.
E-2 data bahasa pemograman tidak valid.
 4. Sistem mengupdate data bahasa pemograman yang dipilih dari database dan menampilkan antarmuka yang menunjukkan bahasa pemograman berhasil dihapus.
 6. Berlanjut ke Basic flow langkah ke-8.

6. Error Flow

- E-1 Data bahasa pemograman tidak valid.
1. Sistem menampilkan pesan peringatan bahwa data bahasa pemograman tidak valid.
 2. Kembali ke Basic flow langkah ke-4.

Program Studi Teknik Informatika	SKPL – SIKSI	53/ 66
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

E-2 data bahasa pemograman tidak valid

1. Sistem menampilkan pesan peringatan bahwa data bahasa pemograman tidak valid.
2. Kembali ke Alternative flow langkah ke-1.

7. PreConditions

1. Use Case Login telah dilakukan.
2. Aktor telah memasuki sistem

8. PostConditions

1. Data ref bahasa pemograman di database telah *ter-update*.

4.1.21 Use case specification: Mengelola Ref Database server

1. Brief Description

Use case ini digunakan oleh aktor untuk mengelola ref database server. Aktor dapat melakukan *view*, *tambah*, dan *edit ref database server*.

2. Primary Actor

1. Administrator

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan *ref database server*.
2. Sistem menampilkan *list database server*.
3. Aktor memilih menambah database server.
A-1 Aktor memilih database server yang ada pada *list database server*.
4. Sistem menampilkan antarmuka untuk memasukkan data database server.
5. Aktor memasukkan data database server.

Program Studi Teknik Informatika	SKPL – SIKSI	54/ 66
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

6. Sistem memeriksa data user valid atau tidak.
E-1 Data database server tidak valid.
7. Sistem menampilkan antarmuka menunjukkan data database server berhasil dimasukkan.
8. Use Case selesai.

5. Alternative Flow

A-1 Aktor memilih database server yang ada pada list database server

1. Sistem menampilkan pilihan untuk melakukan edit.
2. Aktor memasukkan data database server baru yang akan diupdate.
3. Sistem memeriksa apakah data database server valid atau tidak.

E-2 data database server tidak valid.

4. Sistem mengupdate data database server yang dipilih dari database dan menampilkan antarmuka yang menunjukkan database server berhasil dihapus.
5. Berlanjut ke Basic flow langkah ke-8.

6. Error Flow

E-1 Data database server tidak valid.

1. Sistem menampilkan pesan peringatan bahwa data database server tidak valid.
2. Kembali ke Basic flow langkah ke-4.

E-2 data database server tidak valid

1. Sistem menampilkan pesan peringatan bahwa data database server tidak valid.
2. Kembali ke Alternative flow langkah ke-1.

7. PreConditions

1. Use Case Login telah dilakukan.

Program Studi Teknik Informatika	SKPL – SIKSI	55/ 66
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

2. Aktor telah memasuki sistem

8. PostConditions

1. Data ref database server di database telah ter-
update.

4.1.22 Use case specification: Mengelola Ref Jenis APP

1. Brief Description

Use case ini digunakan oleh aktor untuk mengelola ref jenis APP. Aktor dapat melakukan *view, tambah,* dan *edit ref jenis APP.*

2. Primary Actor

1. Administrator

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan *ref jenis APP.*

2. Sistem menampilkan list *jenis APP.*

3. Aktor memilih menambah jenis APP.

A-1 Aktor memilih jenis APP yang ada pada list jenis APP.

4. Sistem menampilkan antarmuka untuk memasukkan data jenis APP.

5. Aktor memasukkan data jenis APP.

6. Sistem memeriksa data user valid atau tidak.

E-1 Data jenis APP tidak valid.

7. Sistem menampilkan antarmuka menunjukkan data jenis APP berhasil dimasukkan.

8. Use Case selesai.

5. Alternative Flow

A-1 Aktor memilih jenis APP yang ada pada list jenis APP

1. Sistem menampilkan pilihan untuk melakukan edit.
2. Aktor memasukkan data jenis APP baru yang akan diupdate.
3. Sistem memeriksa apakah data jenis APP valid atau tidak.

E-2 data jenis APP tidak valid.

4. Sistem mengupdate data jenis APP yang dipilih dari database dan menampilkan antarmuka yang menunjukkan jenis APP berhasil dihapus.
5. Berlanjut ke Basic flow langkah ke-8.

6. Error Flow

E-1 Data jenis APP tidak valid.

1. Sistem menampilkan pesan peringatan bahwa data jenis APP tidak valid.
2. Kembali ke Basic flow langkah ke-4.

E-2 data jenis APP tidak valid

1. Sistem menampilkan pesan peringatan bahwa data jenis APP tidak valid.
2. Kembali ke Alternative flow langkah ke-1.

7. PreConditions

1. Use Case Login telah dilakukan.
2. Aktor telah memasuki sistem

8. PostConditions

1. Data ref jenis APP di database telah ter-update.

Program Studi Teknik Informatika	SKPL – SIKSI	57/ 66
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

4.1.23 Use case specification: Mengelola Ref Model fitur

1. Brief Description

Use case ini digunakan oleh aktor untuk mengelola ref model fitur. Aktor dapat melakukan *view*, *tambah*, dan *edit ref model fitur*.

2. Primary Actor

1. Administrator

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan *ref model fitur*.
2. Sistem menampilkan *list model fitur*.
3. Aktor memilih menambah model fitur.
A-1 Aktor memilih model fitur yang ada pada *list model fitur*.
4. Sistem menampilkan antarmuka untuk memasukkan data model fitur.
5. Aktor memasukkan data model fitur.
6. Sistem memeriksa data user valid atau tidak.
E-1 Data model fitur tidak valid.
7. Sistem menampilkan antarmuka menunjukkan data model fitur berhasil dimasukkan.
8. Use Case selesai.

5. Alternative Flow

- A-1 Aktor memilih model fitur yang ada pada *list model fitur*
1. Sistem menampilkan pilihan untuk melakukan *edit*.

2. Aktor memasukkan data model fitur baru yang akan diupdate.
3. Sistem memeriksa apakah data model fitur valid atau tidak.
E-2 data model fitur tidak valid.
4. Sistem mengupdate data model fitur yang dipilih dari database dan menampilkan antarmuka yang menunjukkan model fitur berhasil dihapus.
5. Berlanjut ke Basic flow langkah ke-8.

6. Error Flow

E-1 Data model fitur tidak valid.

1. Sistem menampilkan pesan peringatan bahwa data model fitur tidak valid.
2. Kembali ke Basic flow langkah ke-4.

E-2 data model fitur tidak valid

1. Sistem menampilkan pesan peringatan bahwa data model fitur tidak valid.
2. Kembali ke Alternative flow langkah ke-1.

7. PreConditions

1. Use Case Login telah dilakukan.
2. Aktor telah memasuki sistem

8. PostConditions

1. Data ref model fitur di database telah ter-update.

4.1.24 Use case specification: Mengelola Ref Sistem operasi

1. Brief Description

Use case ini digunakan oleh aktor untuk mengelola ref sistem operasi. Aktor dapat melakukan *view*, *tambah*, dan *edit ref sistem operasi*.

2. Primary Actor

1. Administrator

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan *ref sistem operasi*.
2. Sistem menampilkan list *sistem operasi*.
3. Aktor memilih menambah sistem operasi.
A-1 Aktor memilih sistem operasi yang ada pada list sistem operasi.
4. Sistem menampilkan antarmuka untuk memasukkan data sistem operasi.
5. Aktor memasukkan data sistem operasi.
6. Sistem memeriksa data user valid atau tidak.
E-1 Data sistem operasi tidak valid.
7. Sistem menampilkan antarmuka menunjukkan data sistem operasi berhasil dimasukkan.
8. Use Case selesai.

5. Alternative Flow

A-1 Aktor memilih sistem operasi yang ada pada list sistem operasi

1. Sistem menampilkan pilihan untuk melakukan edit.

2. Aktor memasukkan data sistem operasi baru yang akan diupdate.
3. Sistem memeriksa apakah data sistem operasi valid atau tidak.
E-2 data sistem operasi tidak valid.
4. Sistem mengupdate data sistem operasi yang dipilih dari database dan menampilkan antarmuka yang menunjukkan sistem operasi berhasil dihapus.
5. Berlanjut ke Basic flow langkah ke-8.

6. Error Flow

E-1 Data sistem operasi tidak valid.

1. Sistem menampilkan pesan peringatan bahwa data sistem operasi tidak valid.
2. Kembali ke Basic flow langkah ke-4.

E-2 data sistem operasi tidak valid

1. Sistem menampilkan pesan peringatan bahwa data sistem operasi tidak valid.
2. Kembali ke Alternative flow langkah ke-1.

7. PreConditions

1. Use Case Login telah dilakukan.
2. Aktor telah memasuki sistem

8. PostConditions

1. Data ref sistem operasi di database telah *ter-update*.

4.1.25 Use case specification: Mengelola Ref Status

1. Brief Description

Use case ini digunakan oleh aktor untuk mengelola ref status. Aktor dapat melakukan *view*, *tambah*, dan *edit ref status*.

Program Studi Teknik Informatika	SKPL – SIKSI	61/ 66
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

2. Primary Actor

1. Administrator

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan *ref status*.
2. Sistem menampilkan *list status*.
3. Aktor memilih menambah status.
A-1 Aktor memilih status yang ada pada *list status*.
4. Sistem menampilkan antarmuka untuk memasukkan data status.
5. Aktor memasukkan data status.
6. Sistem memeriksa data user valid atau tidak.
E-1 Data status tidak valid.
7. Sistem menampilkan antarmuka menunjukkan data status berhasil dimasukkan.
8. Use Case selesai.

5. Alternative Flow

- A-1 Aktor memilih status yang ada pada *list status*
1. Sistem menampilkan pilihan untuk melakukan edit.
 2. Aktor memasukkan data status baru yang akan diupdate.
 3. Sistem memeriksa apakah data status valid atau tidak.
E-2 data status tidak valid.
 4. Sistem mengupdate data status yang dipilih dari database dan menampilkan antarmuka yang menunjukkan status berhasil dihapus.

Program Studi Teknik Informatika	SKPL – SIKSI	62/ 66
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

5. Berlanjut ke Basic flow langkah ke-8.

6. Error Flow

E-1 Data status tidak valid.

1. Sistem menampilkan pesan peringatan bahwa data status tidak valid.
2. Kembali ke Basic flow langkah ke-4.

E-2 data status tidak valid

1. Sistem menampilkan pesan peringatan bahwa data status tidak valid.
2. Kembali ke Alternative flow langkah ke-1.

7. PreConditions

1. Use Case Login telah dilakukan.
2. Aktor telah memasuki sistem

8. PostConditions

1. Data ref status di database telah ter-update.

4.1.26 Use case specification: Mengelola Ref Tools pengembangan

1. Brief Description

Use case ini digunakan oleh aktor untuk mengelola ref tools pengembangan. Aktor dapat melakukan view, tambah, dan edit ref tools pengembangan.

2. Primary Actor

1. Administrator

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan pengelolaan *ref tools pengembangan*.
2. Sistem menampilkan list *tools pengembangan*.
3. Aktor memilih menambah tools pengembangan.

Program Studi Teknik Informatika	SKPL – SIKSI	63/ 66
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

A-1 Aktor memilih tools pengembangan yang ada pada list tools pengembangan.

4. Sistem menampilkan antarmuka untuk memasukkan data tools pengembangan.

5. Aktor memasukkan data tools pengembangan.

6. Sistem memeriksa data user valid atau tidak.

E-1 Data tools pengembangan tidak valid.

7. Sistem menampilkan antarmuka menunjukkan data tools pengembangan berhasil dimasukkan.

8. Use Case selesai.

5. Alternative Flow

A-1 Aktor memilih tools pengembangan yang ada pada list tools pengembangan

1. Sistem menampilkan pilihan untuk melakukan edit.

2. Aktor memasukkan data tools pengembangan baru yang akan diupdate.

3. Sistem memeriksa apakah data tools pengembangan valid atau tidak.

E-2 data tools pengembangan tidak valid.

4. Sistem mengupdate data tools pengembangan yang dipilih dari database dan menampilkan antarmuka yang menunjukkan tools pengembangan berhasil dihapus.

5. Berlanjut ke Basic flow langkah ke-8.

6. Error Flow

E-1 Data tools pengembangan tidak valid.

1. Sistem menampilkan pesan peringatan bahwa data tools pengembangan tidak valid.

2. Kembali ke Basic flow langkah ke-4.

E-2 data tools pengembangan tidak valid

Program Studi Teknik Informatika	SKPL – SIKSI	64/ 66
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

1. Sistem menampilkan pesan peringatan bahwa data tools pengembangan tidak valid.

2. Kembali ke Alternative flow langkah ke-1.

7. PreConditions

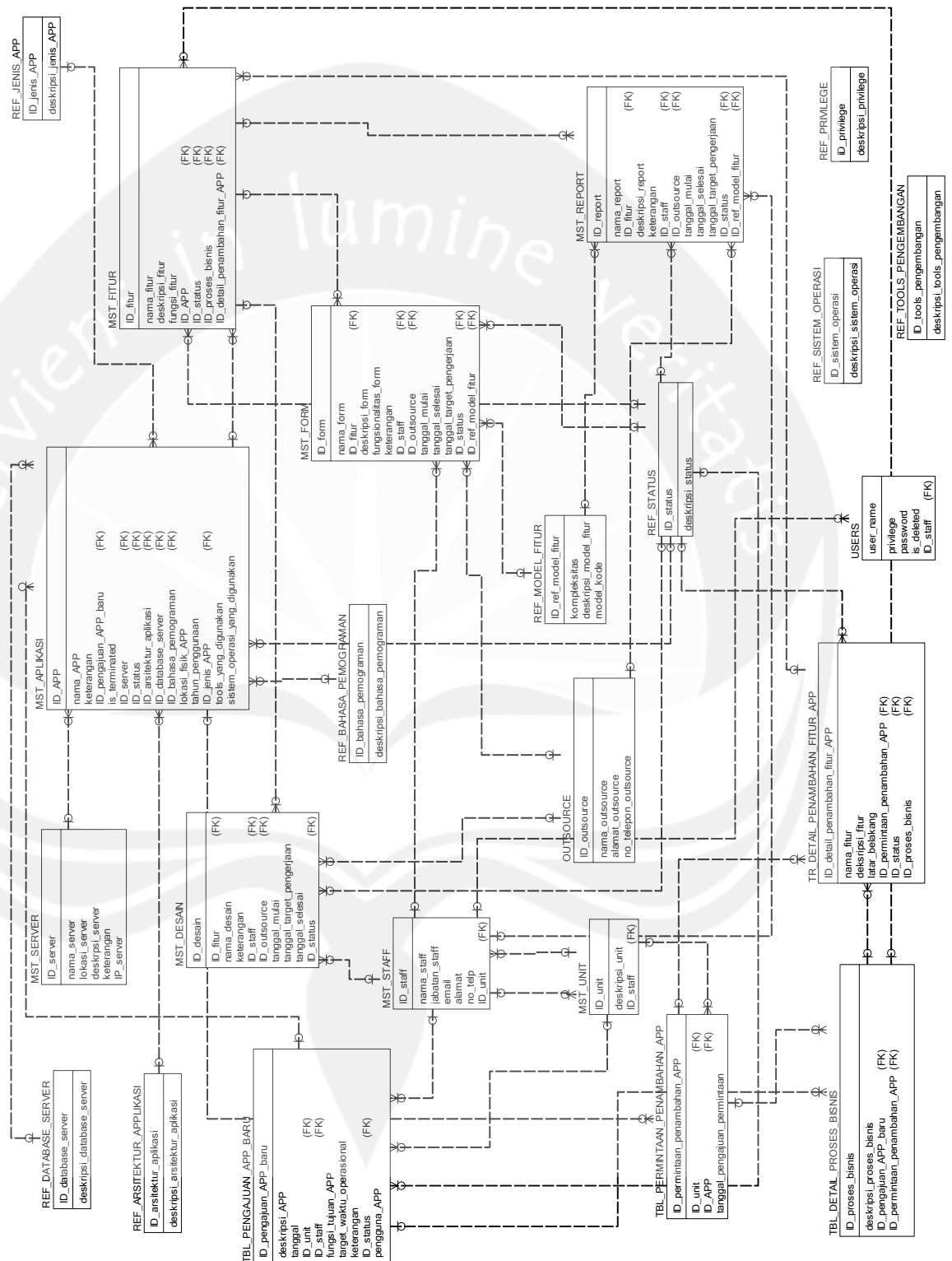
1. Use Case Login telah dilakukan.

2. Aktor telah memasuki sistem

8. PostConditions

1. Data ref tools pengembangan di database telah ter-update.

5. Entity Relationship Diagram (ERD)



Gambar 3. Entity Relationship Diagram

DPPL

DESKRIPSI PERANCANGAN PERANGKAT LUNAK

SIKSI

(Sistem Informasi Kantor Sitem Informasi)

Untuk :


Universitas Atma Jaya Yogyakarta

Dipersiapkan oleh:

Jefri Trio Vember / 08 07 5532

Program Studi Teknik Informatika - Fakultas Teknologi Industri

Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika Fakultas Teknologi Industri	Nomor Dokumen		Halaman
		DPPL-SIKSI		1/100
		Revisi		

DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	
G	

INDEX TGL	-	A	B	C	D	E	F	G
Ditulis oleh	JTV							
Diperiksa oleh	YSP							
Disetujui oleh	IW							

Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi

Daftar Isi

1.	PENDAHULUAN.....	10
1.1	Tujuan.....	10
1.2	Lingkup Masalah.....	10
1.3	Definisi, akronim, dan singkatan.....	10
1.4	Referensi.....	12
2.	PERANCANGAN SISTEM.....	13
2.1	Perancangan Arsitektur.....	13
2.2	Perancangan Rinci.....	14
2.2.1	Sequence Diagram.....	14
2.2.1.1	Login.....	14
2.2.1.2	Pengajuan Pembuatan Aplikasi.....	15
2.2.1.3	Pengajuan Penambahan Fitur.....	15
2.2.1.4	Project Manajemen.....	16
2.2.1.5	Pengelolaan Fitur Project.....	17
2.2.1.6	Pengelolaan Form Aplikasi.....	18
2.2.1.7	Pengelolaan Report Aplikasi.....	18
2.2.1.8	Pengelolaan Desain Aplikasi.....	19
2.2.1.9	Maintenance Aplikasi.....	19
2.2.1.10	Monitoring resource.....	20
2.2.1.11	Cetak laporan project manager.....	20
2.2.1.12	Laporan perkembangan project.....	21
2.2.1.13	Penugasan Project.....	21
2.2.1.14	Task.....	22
2.2.1.15	Cetak Surat CCP.....	23
2.2.1.16	Laporan sistem informasi.....	23
2.2.1.17	Inventarisasi Sistem Informasi.....	24
2.2.1.18	Kelola User.....	25
2.2.1.19	Kelola ref arsitektur aplikasi.....	26
2.2.1.20	Kelola Ref Bahasa Pemrograman.....	27
2.2.1.21	Kelola Ref Database Server.....	28
2.2.1.22	Kelola Jenis Aplikasi.....	29
2.2.1.23	Kelola Ref Model Fitur.....	30
2.2.1.24	Kelola ref sistem operasi.....	31
2.2.1.25	Kelola Ref Status.....	32
2.2.1.26	Kelola Ref Tools Pengembangan.....	33
2.2.2	Class Diagram.....	34
2.2.3	Class Diagram Specific Descriptions.....	38
2.2.3.1	Spesific Design Class adminJs.....	38
2.2.3.2	Spesific Design Class LoginUI.....	38
2.2.3.3	Spesific Design Class PermohonanJS..	39
2.2.3.4	Spesific Design Class penambahanFiturJS	39
2.2.3.5	Spesific Design Class LaporanSistemInformasiJS	41
2.2.3.6	Spesific Design Class maintenancaAplikasiJS	41

2.2.3.7	Spesific Design Class projectManajemenJS	42
2.2.3.8	Spesific Design Class InventarisAplikasiJS	43
2.2.3.9	Spesific Design Class laporanPerkembanganProjectJS	44
2.2.3.10	Spesific Design Class pengelolaanFiturJS	45
2.2.3.11	Spesific Design Class penugasanProjectJS	45
2.2.3.12	Spesific Design Class taskProjectJS ..	46
2.2.3.13	Spesific Design Class pengelolaanDesainJS	47
2.2.3.14	Spesific Design Class pengelolaanFormJS	47
2.2.3.15	Spesific Design Class pengelolaanReportJS	48
2.2.3.16	Spesific Design Class resourceJS	48
2.2.3.17	Spesific Design Class CetakLaporanManajerUI	48
2.2.3.18	Spesific Design Class CetakSuratCCPUI	49
2.2.3.19	Spesific Design Class adminWS	49
2.2.3.20	Spesific Design Class LoginManager ..	50
2.2.3.21	Spesific Design Class UserWS	50
2.2.3.22	Spesific Design Class PengajuanAPPBaruWS	51
2.2.3.23	Spesific Design Class detailProsesBisnisWS	52
2.2.3.24	Spesific Design Class LaporanSistemInformasiWS	52
2.2.3.25	Spesific Design Class PenambahanFiturWS	53
2.2.3.26	Spesific Design Class DetailPenambahanFiturWS	53
2.2.3.27	Spesific Design Class MaintenanceAplikasiWS	54
2.2.3.28	Spesific Design Class InventarisAplikasiWS	54
2.2.3.29	Spesific Design Class FiturWS	54
2.2.3.30	Spesific Design Class detailAPPWS ...	55
2.2.3.31	Spesific Design Class AplikasiWS	56
2.2.3.32	Spesific Design Class FormWS	57
2.2.3.33	Spesific Design Class ReportWS	58
2.2.3.34	Spesific Design Class DesainWS	59
2.2.3.35	Spesific Design Class StaffWS	60
2.2.3.36	Spesific Design Class OutsourceWS ...	60
2.2.3.37	Spesific Design Class ResourceWS	60
2.2.3.38	Spesific Design Class LaporanProyekManajerControl	61
2.2.3.39	Spesific Design Class cetakSuratCCPManager	61
2.2.3.40	Spesific Design Class ToolsDAO	61
2.2.3.41	Spesific Design Class ArsitekturAplikasiDAO	62

2.2.3.42	Spesific Design Class BahasaPemrogramanDAO	62
2.2.3.43	Spesific Design Class SistemOperasiDAO	63
2.2.3.44	Spesific Design Class DatabaseServerDAO	63
2.2.3.45	Spesific Design Class JenisAPPDAO...	64
2.2.3.46	Spesific Design Class ModelFiturDAO.	64
2.2.3.47	Spesific Design Class ModelFiturDAO.	65
2.2.3.48	Spesific Design Class UserDAO.....	65
2.2.3.49	Spesific Design Class PengajuanAPPBaruDAO	66
2.2.3.50	Spesific Design Class PenambahanFiturDAO	67
2.2.3.51	Spesific Design Class DetailPenambahanFiturDAO	68
2.2.3.52	Spesific Design Class MaintenanceAplikasiDAO	68
2.2.3.53	Spesific Design Class FiturDAO.....	69
2.2.3.54	Spesific Design Class detailAppDAO..	70
2.2.3.55	Spesific Design Class AplikasiDAO...	71
2.2.3.56	Spesific Design Class FormDAO.....	71
2.2.3.57	Spesific Design Class ReportDAO.....	73
2.2.3.58	Spesific Design Class DesainDAO.....	74
2.2.3.59	Spesific Design Class StaffDAO.....	75
2.2.3.60	Spesific Design Class ResourceDAO...	76
2.2.3.61	Spesific Design Class PrintLaporanManajerDAO	76
2.2.3.62	Spesific Design Class cetakSuratCCPDAO	77
2.2.3.63	Spesific Design Class OutsourceDAO..	77
3.	PERANCANGAN DATA.....	78
3.1	Dekomposisi Data	78
3.1.1	Deskripsi Entitas Aplikasi.....	78
3.1.2	Deskripsi Entitas Desain.....	79
3.1.3	Deskripsi Entitas Fitur.....	79
3.1.4	Deskripsi Entitas Form.....	80
3.1.5	Deskripsi Entitas Outsource.....	81
3.1.6	Deskripsi Entitas Report.....	81
3.1.7	Deskripsi entitas server.....	82
3.1.8	Deskripsi Entitas Staff.....	82
3.1.9	Deskripsi Entitas Unit.....	82
3.1.10	Deskripsi Entitas Arsitektur Aplikasi.....	83
3.1.11	Deskripsi Entitas Bahasa Pemrograman.....	83
3.1.12	Deskripsi Entitas Database Server.....	83
3.1.13	Deskripsi Entitas Jenis Aplikasi.....	83
3.1.14	Deskripsi Entitas Model Fitur.....	83
3.1.15	Deskripsi Entitas Privilege.....	84
3.1.16	Deskripsi Entitas Sistem Operasi.....	84
3.1.17	Deskripsi Entitas Bahasa Pemrograman.....	84
3.1.18	Deskripsi Entitas Tools Pengembangan.....	84
3.1.19	Deskripsi Entitas Pengajuan Aplikasi baru..	85

3.1.20	Deskripsi Entitas Permintaan Penambahan Fitur Aplikasi.....	85
3.1.21	Deskripsi Entitas Users.....	86
3.1.22	Deskripsi Entitas Detail Penambahan Fitur..	86
3.1.23	Deskripsi Entitas Detail Proses Bisnis.....	87
3.1	Physic Data Model	89
4.	PERANCANGAN ANTARMUKA.....	90
4.1	Login	90
4.2	Laporan Sistem Informasi	91
4.3	Inventaris Sistem Informasi	91
4.4	Detail Inventaris Sistem Informasi.....	92
4.5	Pengajuan Permohonan Pembuatan Aplikasi Baru	93
4.6	Form Pengajuan Pembuatan Aplikasi Baru ...	94
4.7	Pengajuan Penambahan Fitur	95
4.8	Laporan Perkembangan Project	95
4.9	Project Management	96
4.10	Penugasan Project - Penugasan Project	97
4.11	Task - On Progress Task	98
4.12	Task - Complete Task	99
4.13	Cetak Surat CPP	100

Daftar Gambar

Gambar 1	Perancangan Arsitektur.....	13
Gambar 2.	Sequence Diagram: Login.....	14
Gambar 3.	Sequence Diagram: Pengajuan Permohonan Pembuatan Aplikasi Baru.....	15
Gambar 4.	Sequence Diagram: Pengajuan Penambahan Fitur...	15
Gambar 5.	Sequence Diagram: Approve Permohonan Pembuatan Aplikasi.....	16
Gambar 6.	Sequence Diagram: Tambah Ffitur Aplikasi.....	16
Gambar 7.	Sequence Diagram: Pengelolaan Fitur Project....	17
Gambar 8.	Sequence Diagram: Pengelolaan Form Aplikasi....	18
Gambar 9.	Sequence Diagram: Pengelolaan Report Aplikasi..	18
Gambar 10.	Sequence Diagram: Pengelolaan Desain Aplikasi .	19
Gambar 11.	Sequence Diagram: Maintenance Aplikasi	19
Gambar 12.	Sequence Diagram: Monitoring Resource	20
Gambar 13.	Sequence Diagram: Cetak Laporan Project Manager	20
Gambar 14.	Sequence Diagram: Laporan Perkembangan Project	21
Gambar 15.	Sequence Diagram: Penugasan Project	21
Gambar 16.	Sequence Diagram : Task	22
Gambar 17.	Sequence Diagram : Cetak Surat CCP	23
Gambar 18.	Sequence Diagram: Laporan Sistem Informasi	23
Gambar 19.	Sequence Diagram: Inventarisasi Sistem Informasi	24
Gambar 20.	Sequence Diagram: Kelola User	25
Gambar 21.	Sequence Diagram: Kelola Ref Arsitektur	26
Gambar 22.	Sequence Diagram: Kelola ref Bahasa Pemrograman	27
Gambar 23.	Sequence Diagram: Kelola Ref Database Server ..	28
Gambar 24.	Sequence Diagram: Kelola Ref Jenis Aplikasi ...	29
Gambar 25.	Sequence Diagram: Kelola Ref Model Fitur	30
Gambar 26.	Sequence Diagram: Kelola Ref Sistem Operasi ...	31
Gambar 27.	Sequence Diagram: Kelola Ref Status	32
Gambar 28.	Sequence Diagram: Kelola Ref Tools Pengembangan	33
Gambar 29.	Class Diagram	37
Gambar 30.	Physic Data Model	89

Gambar 31. Rancangan Antarmuka Login	90
Gambar 32. Rancangan Antarmuka Laporan Sistem Informasi ..	91
Gambar 33. Rancangan Laporan Sistem informasi	91
Gambar 34. Rancangan Antarmuka Detail Inventaris Sistem Informasi.....	92
Gambar 35. Rancangan Antarmuka Pengajuan Permohonan Pembuatan Aplikasi Baru.....	93
Gambar 36. Rancangan Antarmuka Form Pengajuan Pembuatan Aplikasi Baru.....	94
Gambar 37. Rancangan Antarmuka Pengajuan Penambahan Fitur	95
Gambar 38. Rancangan Antarmuka Laporan Perkembangan Project.....	95
Gambar 39. Rancangan Antarmuka Project Management	96
Gambar 40. Rancangan Antarmuka Penugasan Project	97
Gambar 41. Rancangan Antarmuka Task - On Progress Task ...	98
Gambar 42. Rancangan Antarmuka Task - Complete Task	99

1. Pendahuluan

1.1 Tujuan

Dokumen Deskripsi Perancangan Perangkat Lunak (DPPL) bertujuan untuk mendefinisikan perancangan perangkat lunak yang akan dikembangkan. Dokumen DPPL tersebut digunakan oleh pengembang perangkat lunak sebagai acuan untuk implementasi pada tahap selanjutnya.

1.2 Lingkup Masalah

Perangkat Lunak SIKSI dikembangkan dengan tujuan untuk :

1. Menangani pengajuan permohonan pembuatan aplikasi dalam lingkup Universitas Atma Jaya Yogyakarta.
2. Menangani pengajuan penambahan fitur aplikasi.
3. Menangani proses persetujuan permohonan pembuatan aplikasi.
4. Menangani proses pengelolaan permohonan pembuatan aplikasi.
5. Menangani proses pengelolaan penambahan fitur aplikasi.
6. Menyediakan pemantauan kinerja staff KSI.
7. Menyediakan laporan untuk proyek manajer.
8. Menyediakan notifikasi penugasan proyek untuk staff IT.
9. Menyediakan laporan CCP untuk staff IT.
10. Menangani proses inventarisasi sistem/aplikasi dan menyediakan laporan deskripsi sistem/aplikasi.

1.3 Definisi, akronim, dan singkatan

Daftar definisi akronim dan singkatan :

Program Studi Teknik Informatika	DPPL – SIKSI	10/ 100
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Keyword/Phrase	Definisi
SKPL	Deskripsi Perancangan Perangkat Lunak disebut juga Software Design Description (SDD) merupakan deskripsi dari perancangan produk/perangkat lunak yang akan dikembangkan.
SIKSI	Sistem Informasi Kantor Sistem Informasi
KSI	Kantor Sistem Informasi
Internet	Internet merupakan istilah umum yang dipakai untuk menunjuk <i>Network</i> global yang terdiri dari komputer dan layanan servis dengan sekitar 30 sampai 50 juta pemakai komputer dan puluhan layanan informasi termasuk e-mail, FTP, dan <i>World Wide Web</i> .
Server	Komputer yang menyediakan sumber daya bagi klien yang terhubung melalui jaringan.
<i>Client</i>	Komputer yang menghubungi <i>server</i> melalui jaringan
<i>Web based</i>	Menyatakan aplikasi yang dalam penggunaannya pada komputer client dapat diakses mempergunakan web browser tanpa memerlukan proses instalasi terlebih dahulu.
IIS	Internet Information Services HTTP web server yang digunakan dalam

	sistem operasi server Windows.
<i>Privilege</i>	Menyatakan peran pengguna di dalam sistem yang menyangkut hak akses pengguna tersebut.
<i>CCP</i>	<i>Curriculum Credit Point</i>

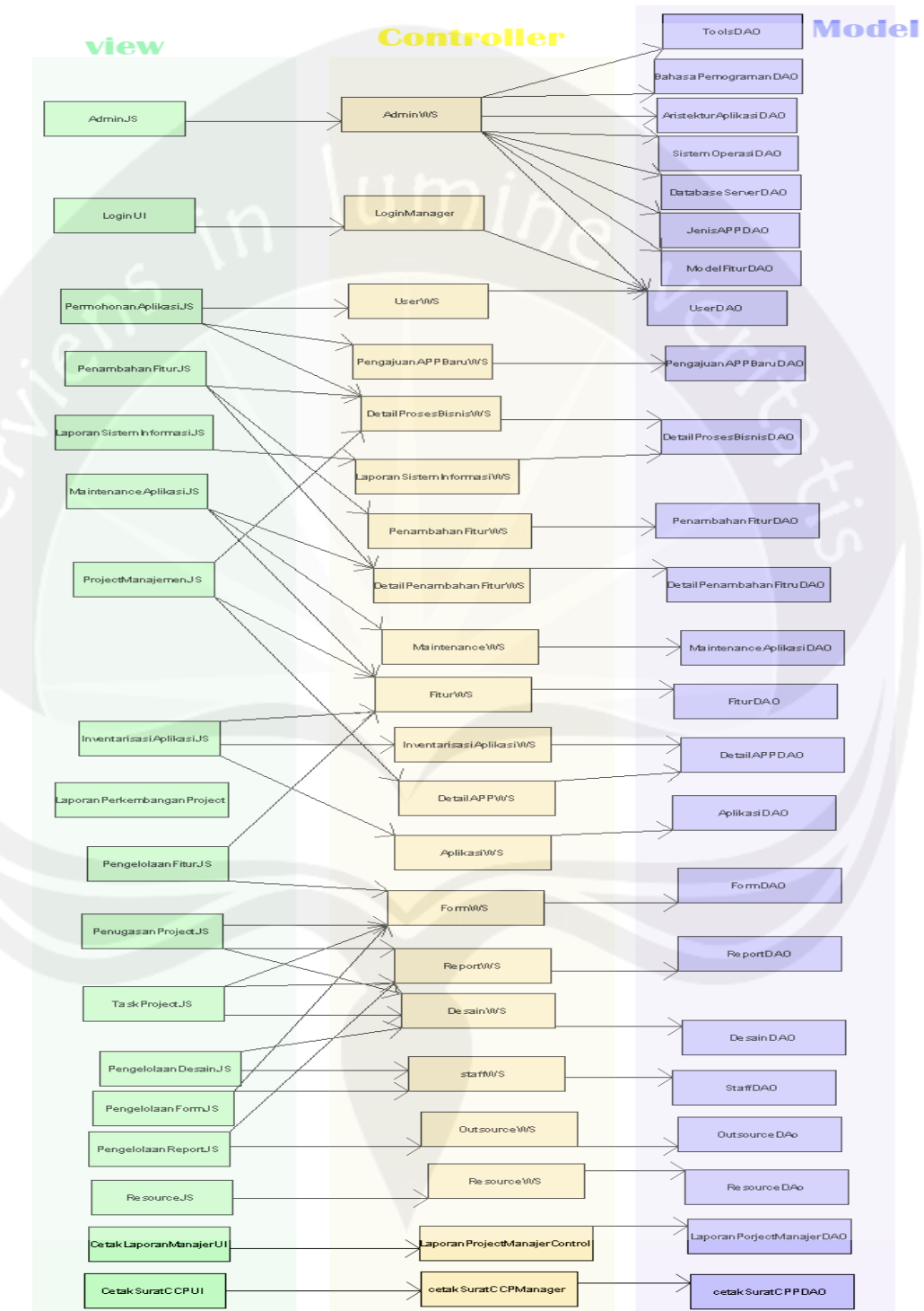
1.4 Referensi

Referensi yang digunakan pada perangkat lunak tersebut adalah:

1. Bennet Simon, McRobb Steve, Farmer Ray, *Object-Oriented Sistem Analysis and Design Using UML*, McGraw-Hill Companies, 2002.
2. Jefri Trio Vember, *Spesifikasi Kebutuhan Perangkat Lunak SIKSI UAJY*, Universitas Atma Jaya Yogyakarta, 2012.

2. Perancangan Sistem

2.1 Perancangan Arsitektur

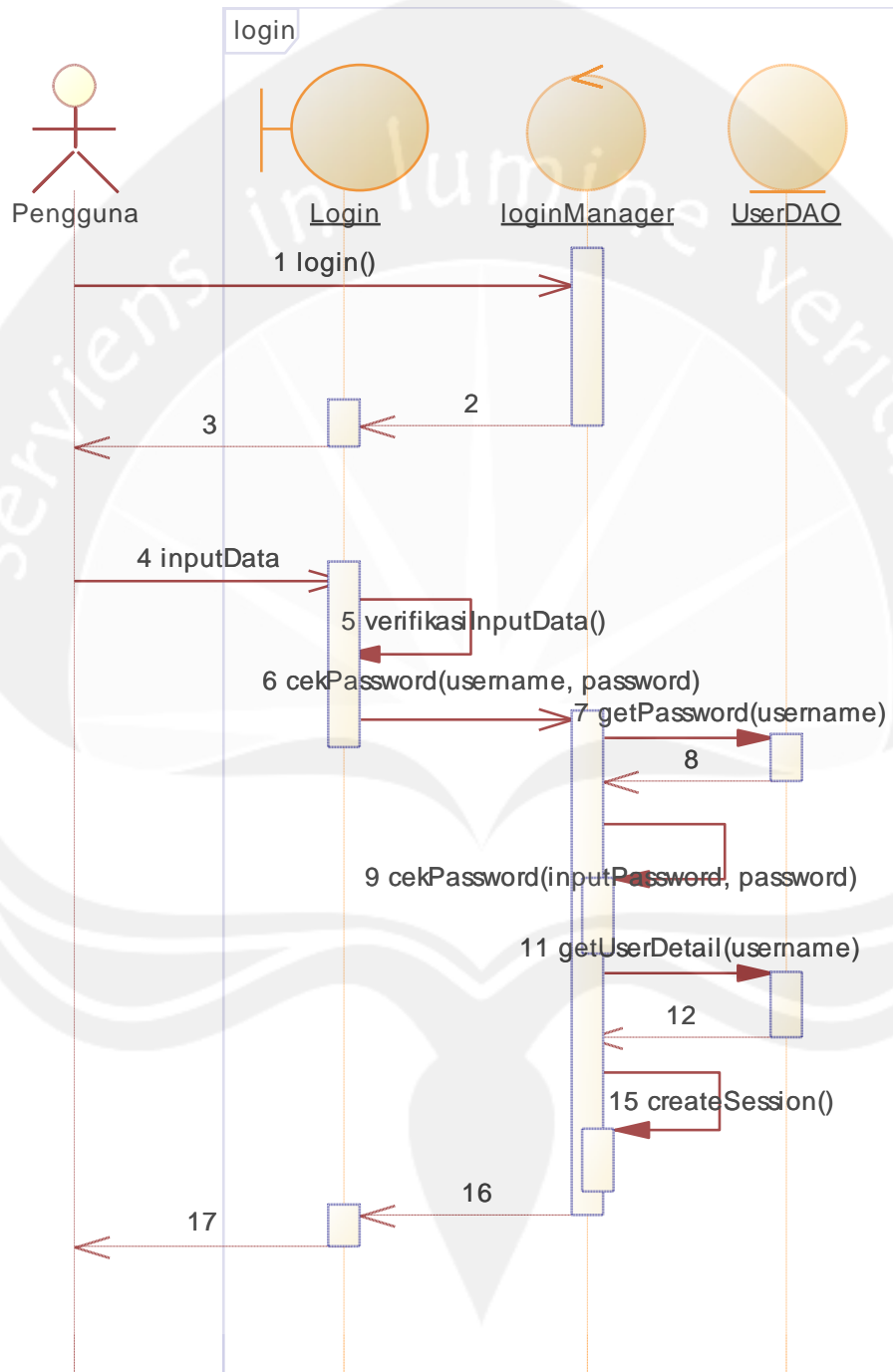


Gambar 1 Perancangan Arsitektur

2.2 Perancangan Rinci

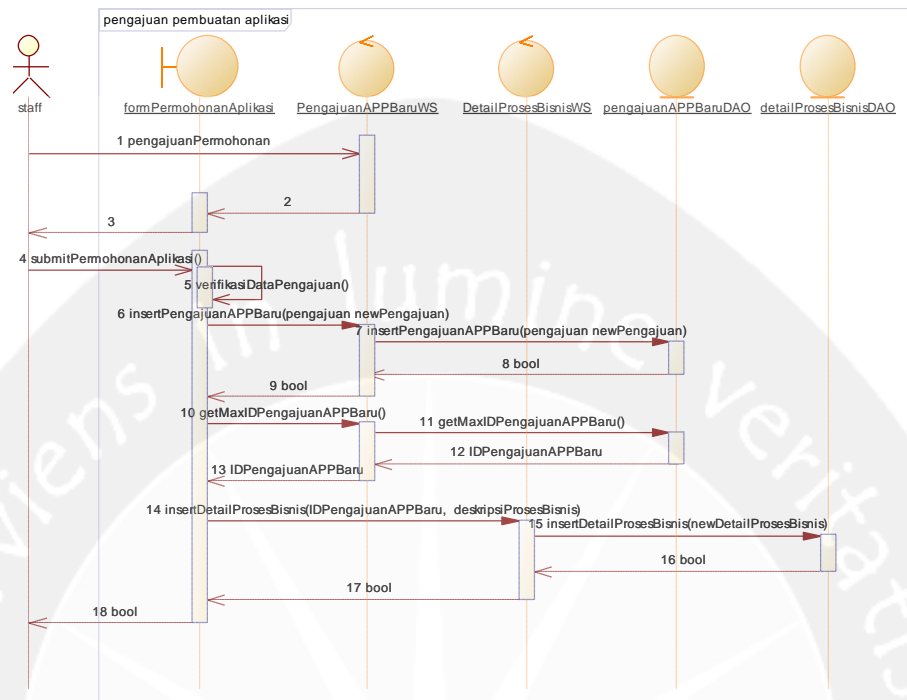
2.2.1 Sequence Diagram

2.2.1.1 Login



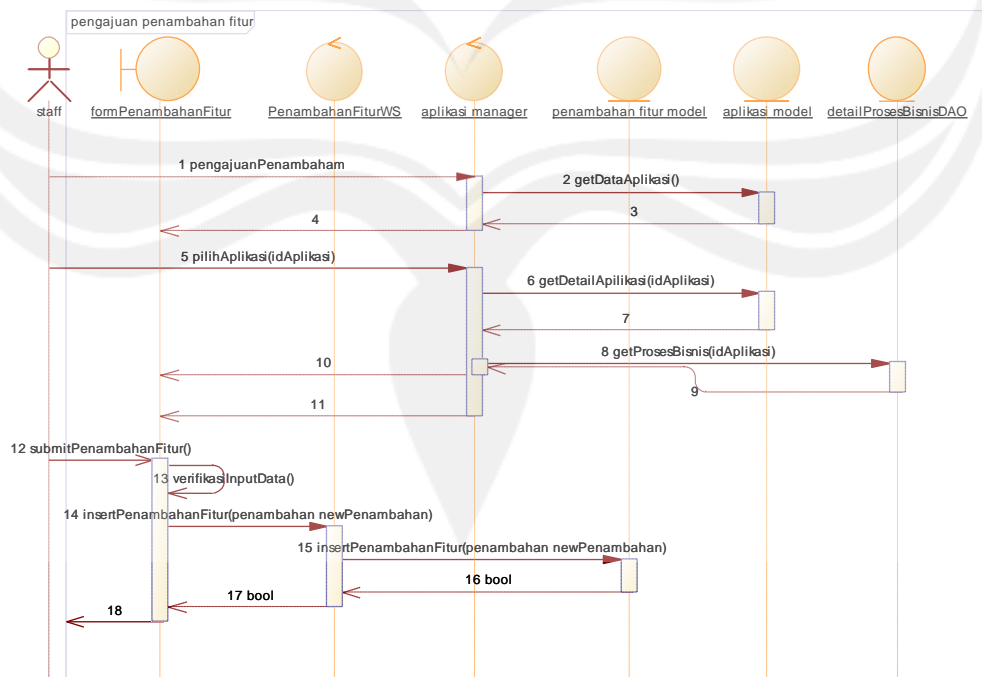
Gambar 2. Sequence Diagram: Login

2.2.1.2 Pengajuan Pembuatan Aplikasi



Gambar 3. Sequence Diagram: Pengajuan Permohonan Pembuatan Aplikasi Baru

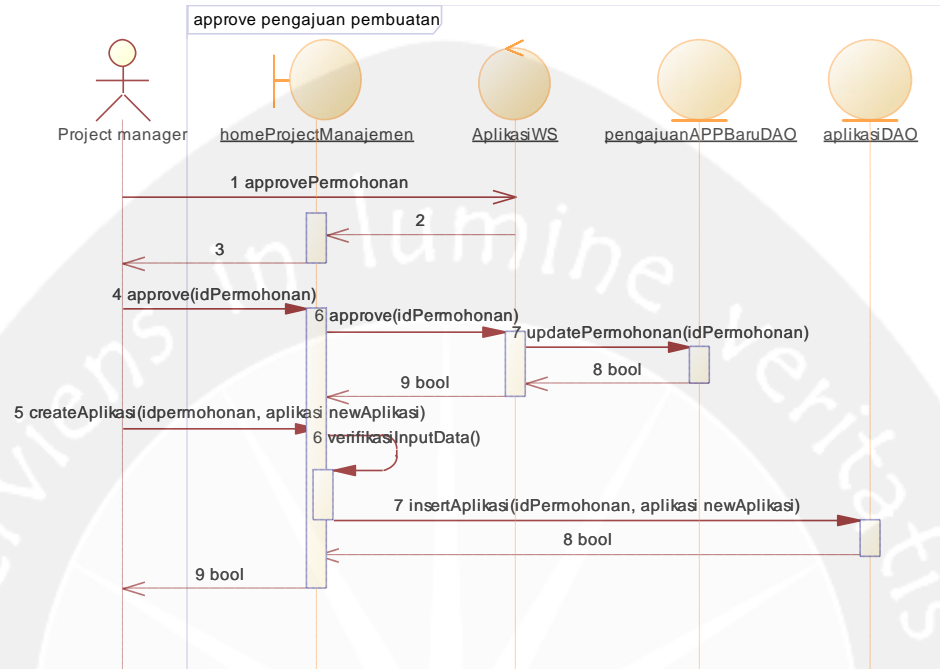
2.2.1.3 Pengajuan Penambahan Fitur



Gambar 4. Sequence Diagram: Pengajuan Penambahan Fitur

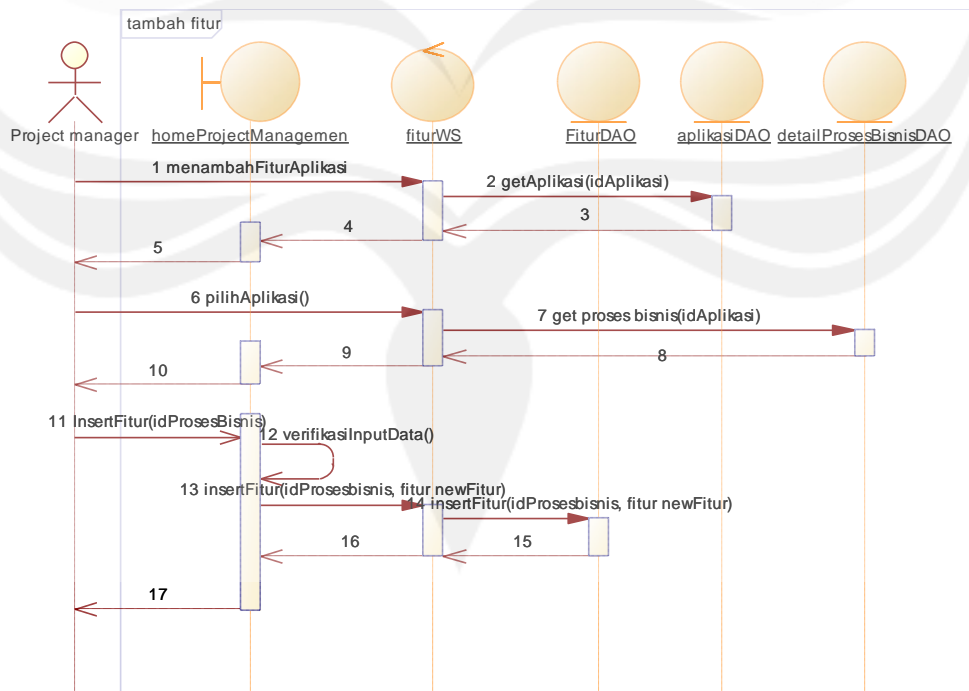
2.2.1.4 Project Manajemen

2.2.1.4.1 Approve Permohonan



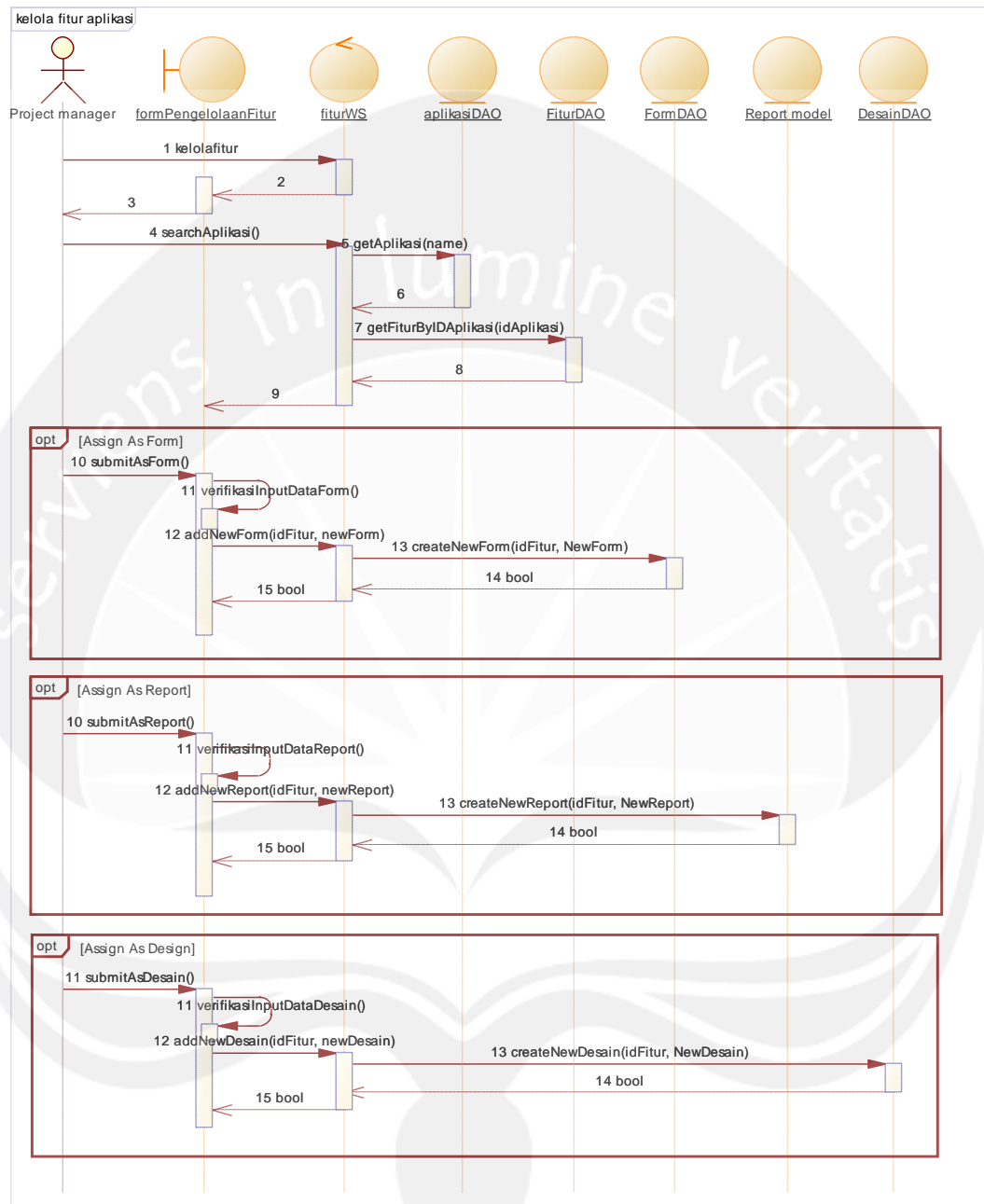
Gambar 5. Sequence Diagram: Approve Permohonan Pembuatan Aplikasi

2.2.1.4.2 Tambah Fitur



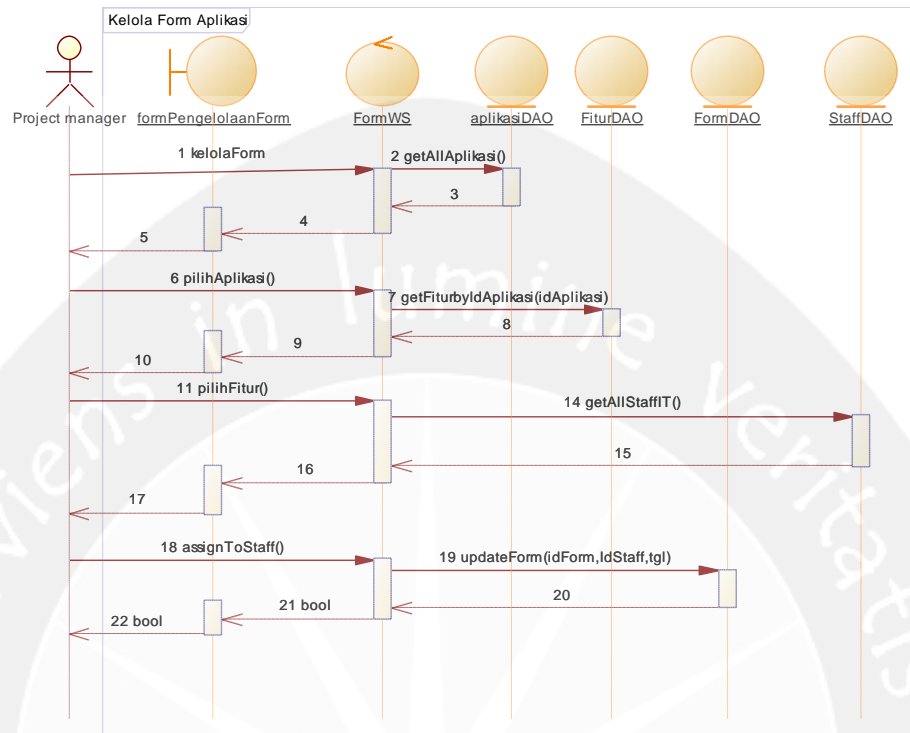
Gambar 6. Sequence Diagram: Tambah Ffitur Aplikasi

2.2.1.5 Pengelolaan Fitur Project



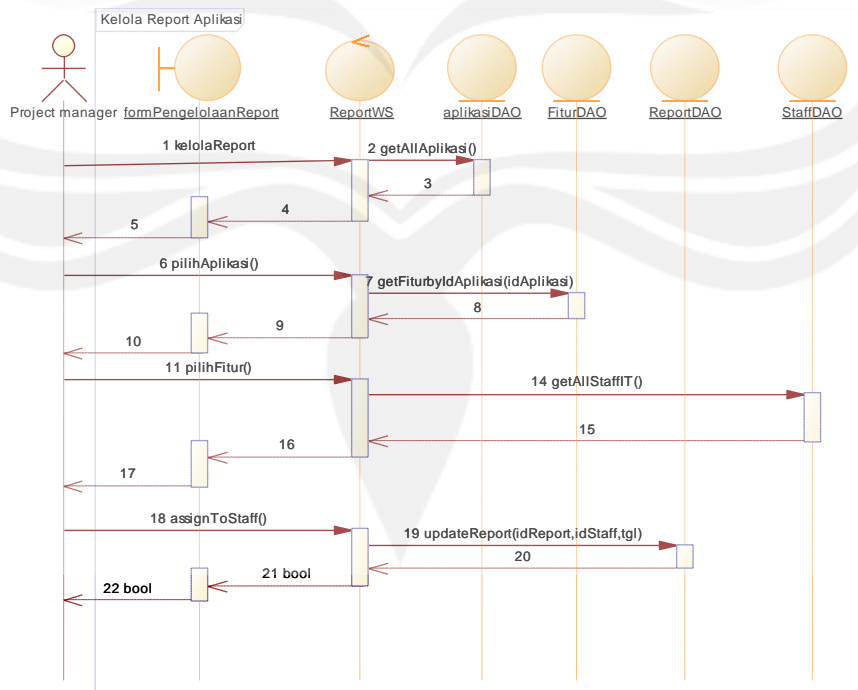
Gambar 7. Sequence Diagram: Pengelolaan Fitur Project

2.2.1.6 Pengelolaan Form Aplikasi



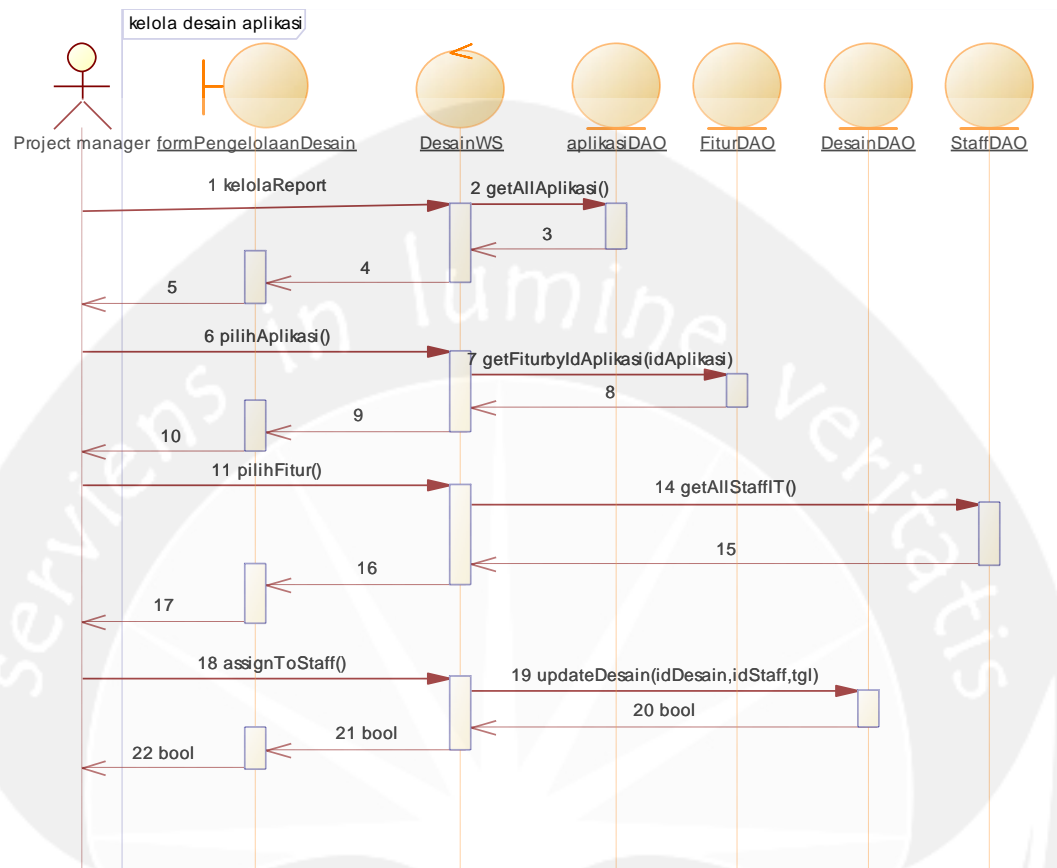
Gambar 8. Sequence Diagram: Pengelolaan Form Aplikasi

2.2.1.7 Pengelolaan Report Aplikasi



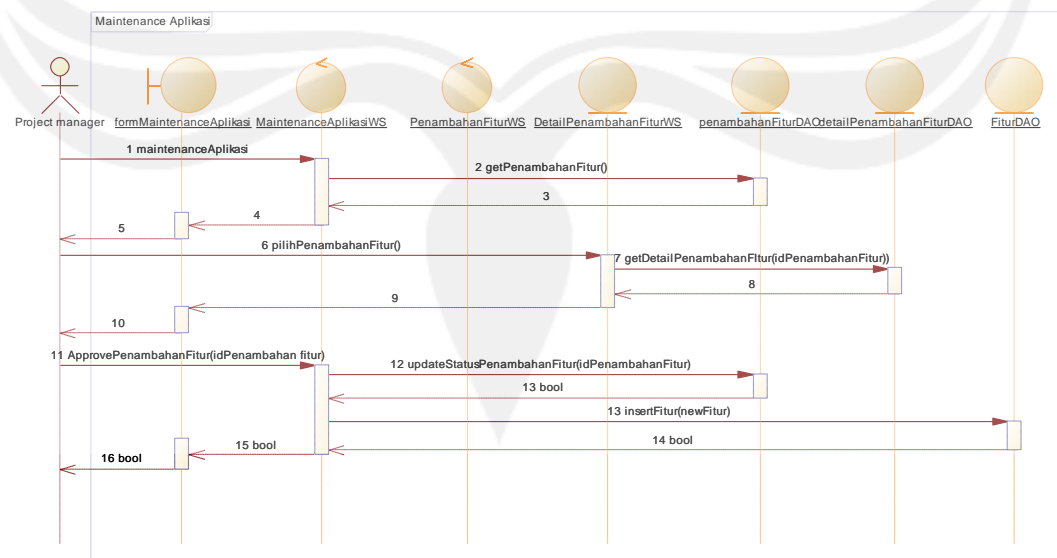
Gambar 9. Sequence Diagram: Pengelolaan Report Aplikasi

2.2.1.8 Pengelolaan Desain Aplikasi



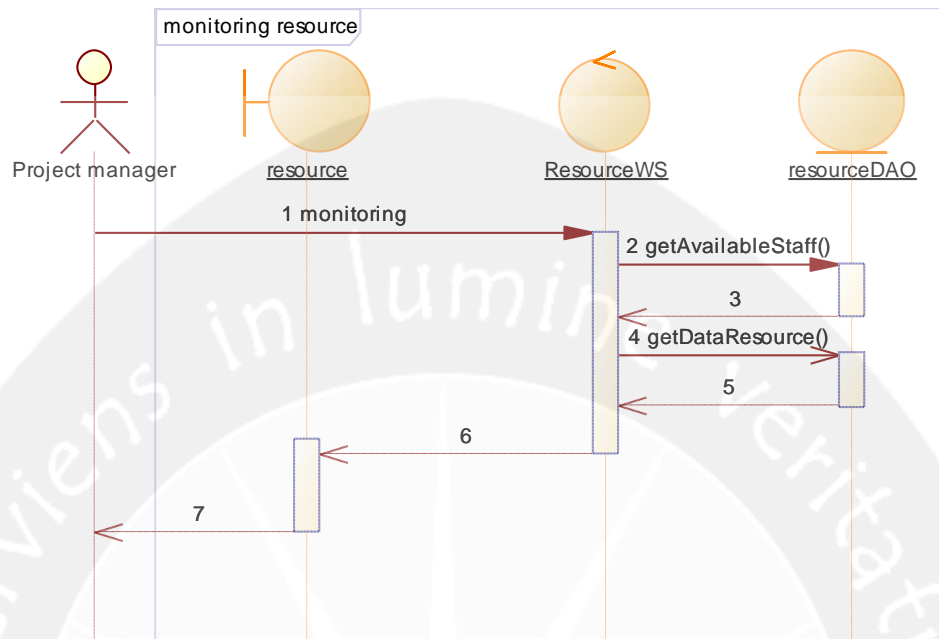
Gambar 10. Sequence Diagram: Pengelolaan Desain Aplikasi

2.2.1.9 Maintenance Aplikasi



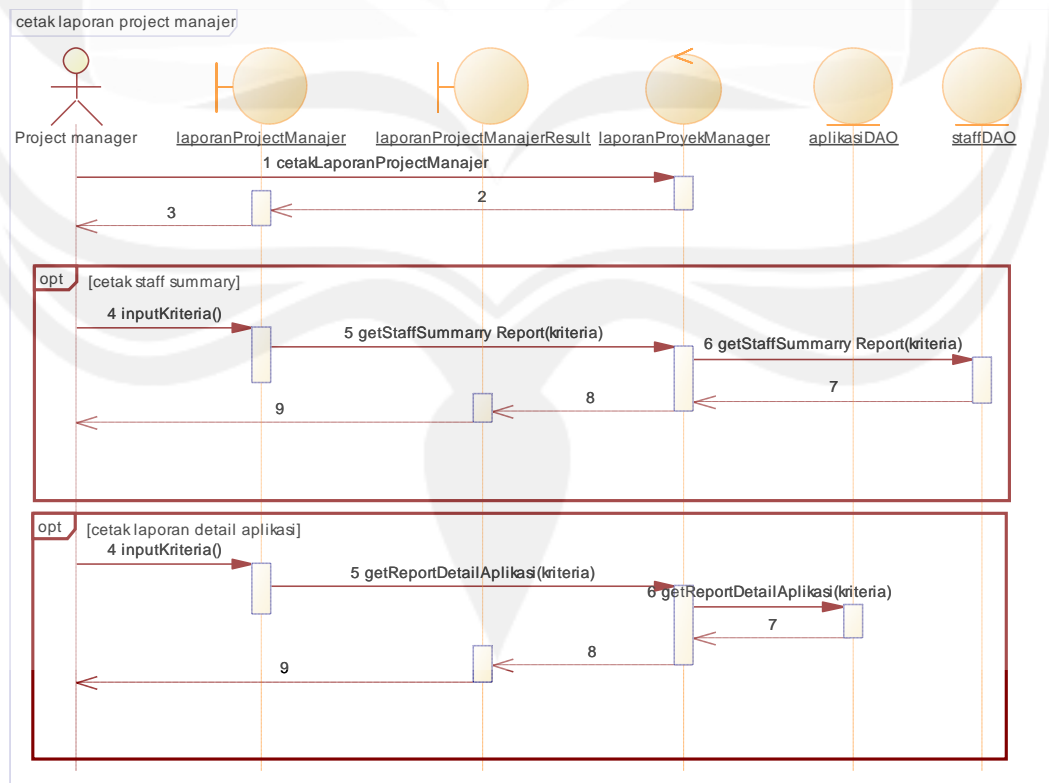
Gambar 11. Sequence Diagram: Maintenance Aplikasi

2.2.1.10 Monitoring resource



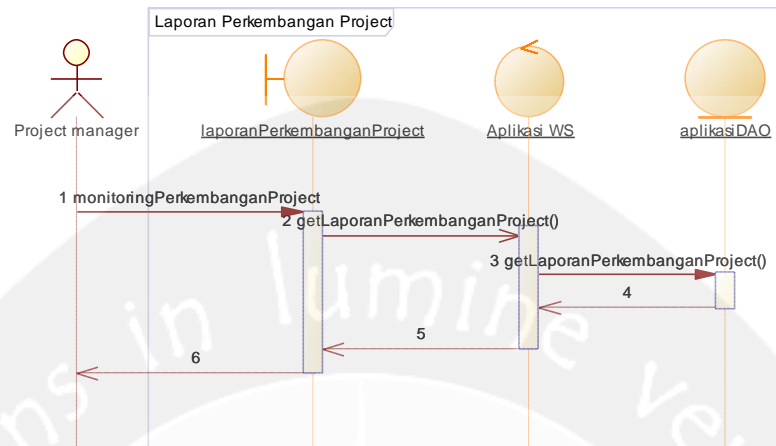
Gambar 12. Sequence Diagram: Monitoring Resource

2.2.1.11 Cetak laporan project manager



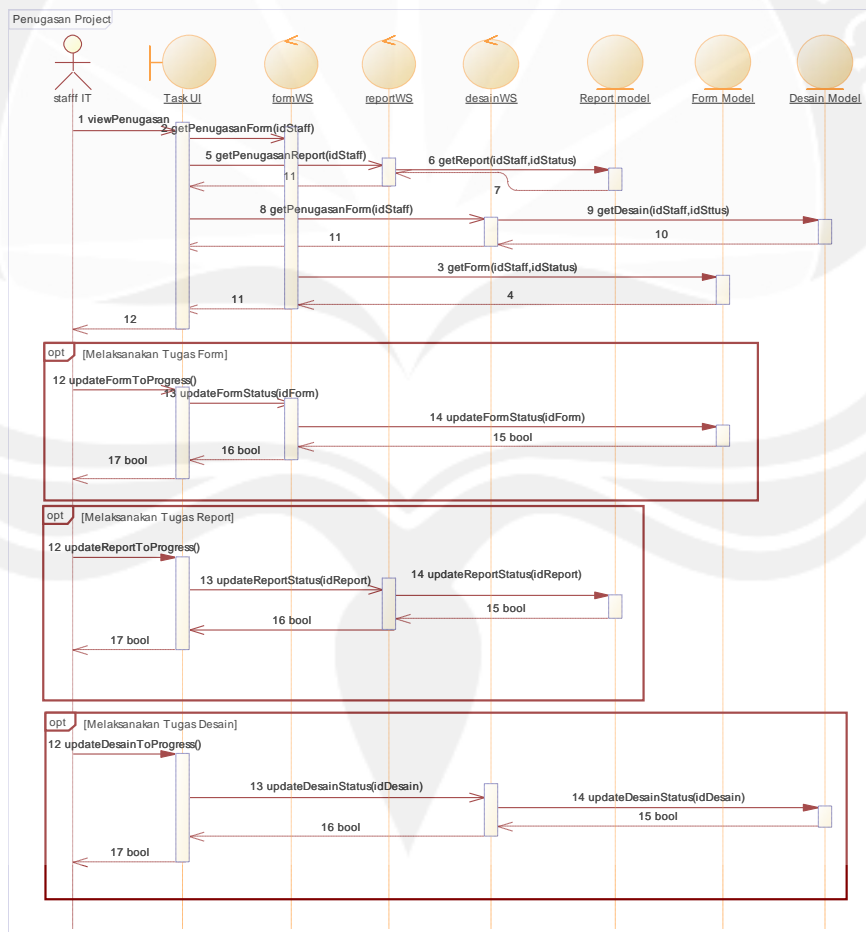
Gambar 13. Sequence Diagram: Cetak Laporan Project Manager

2.2.1.12 Laporan perkembangan project



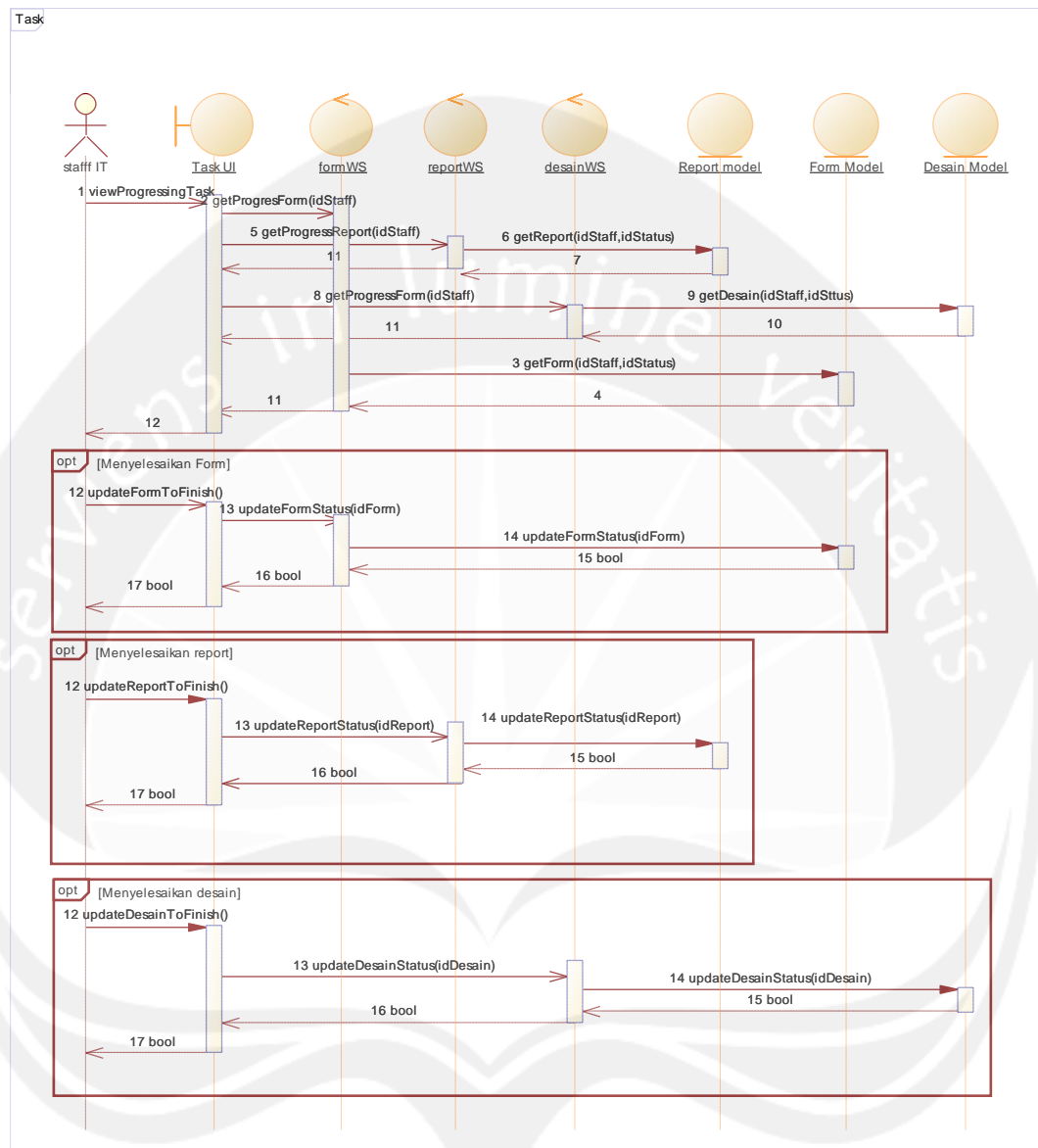
Gambar 14. Sequence Diagram: Laporan Perkembangan Project

2.2.1.13 Penugasan Project



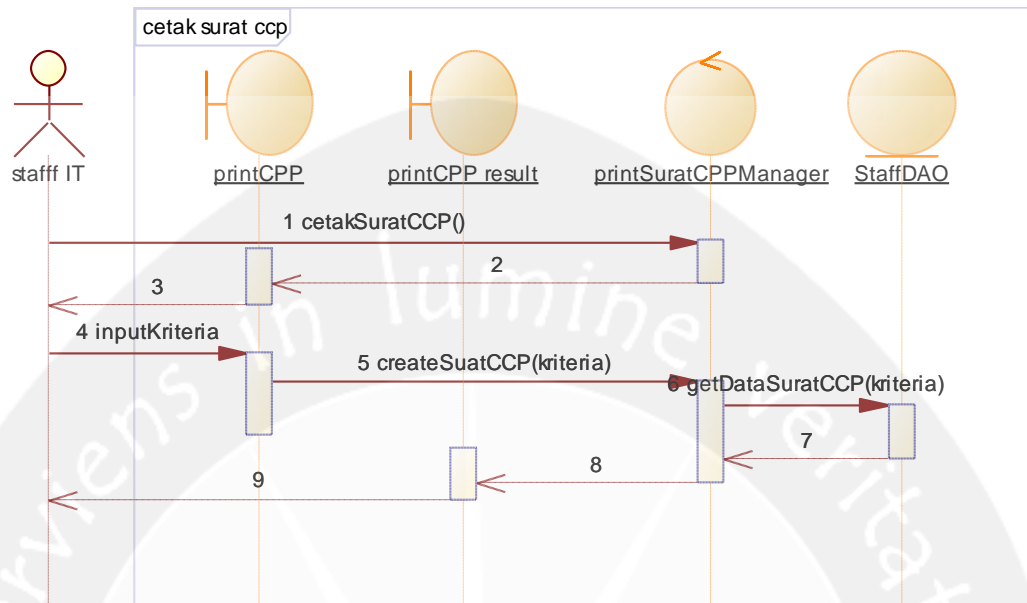
Gambar 15. Sequence Diagram: Penugasan Project

2.2.1.14 Task



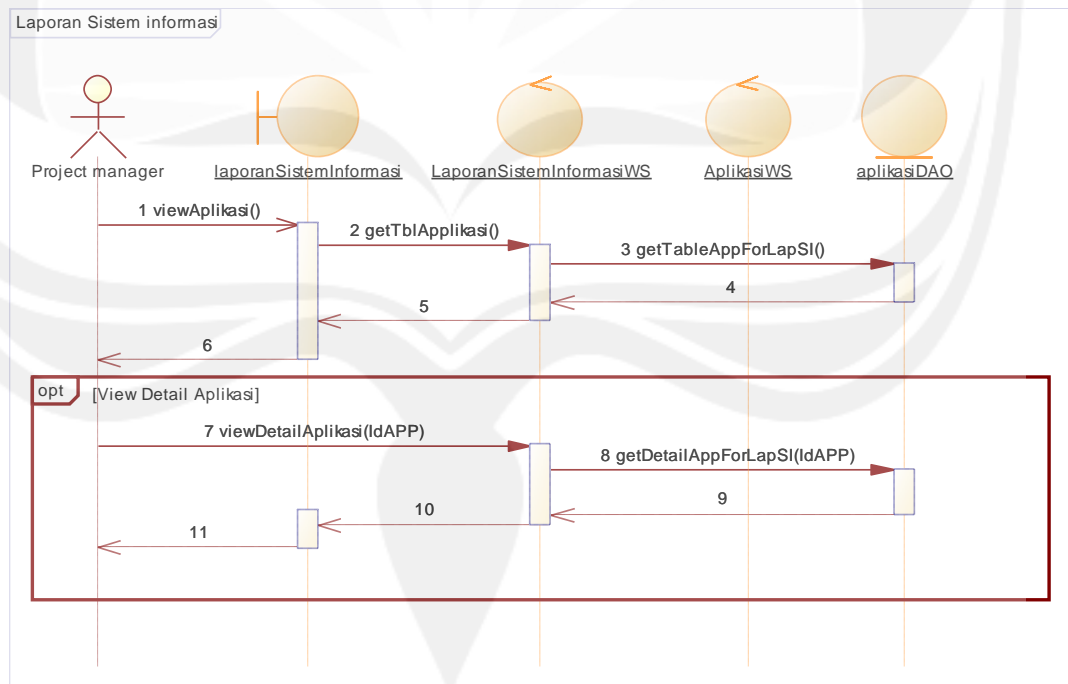
Gambar 16. Sequence Diagram : Task

2.2.1.15 Cetak Surat CCP



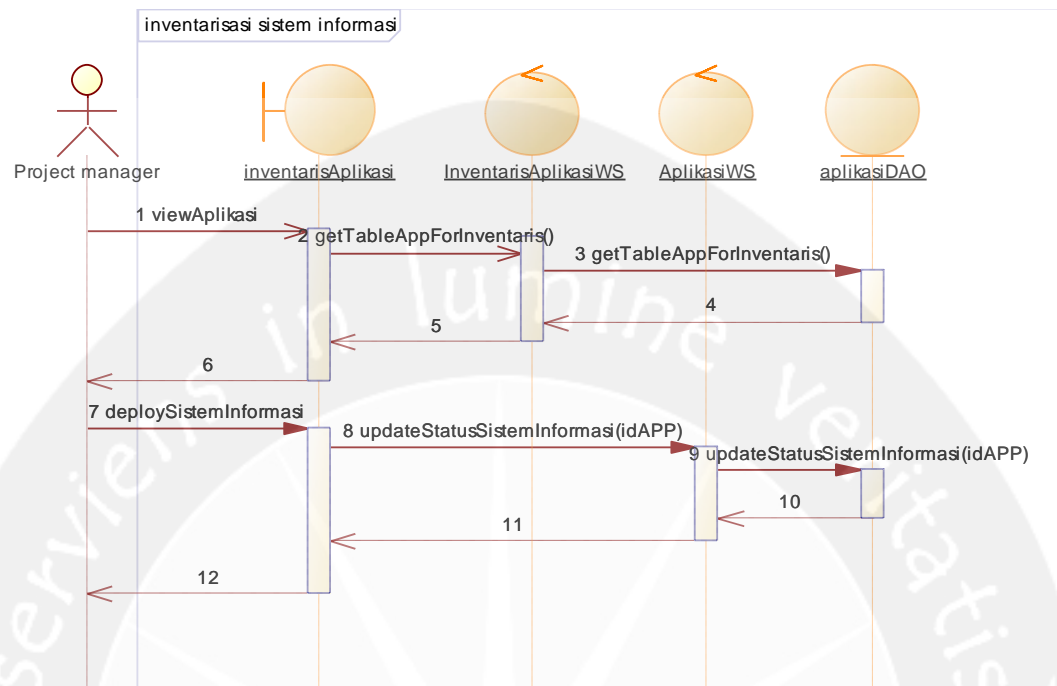
Gambar 17. Sequence Diagram : Cetak Surat CCP

2.2.1.16 Laporan sistem informasi



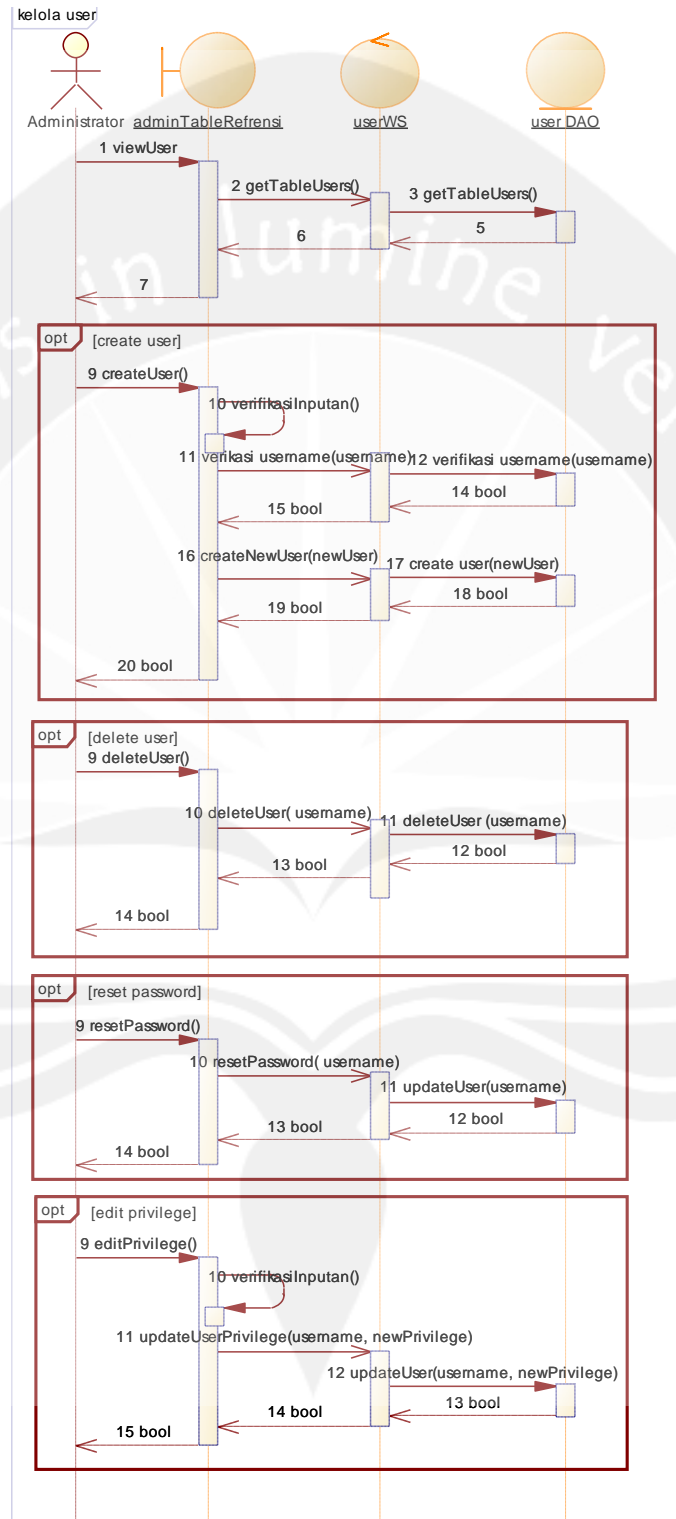
Gambar 18. Sequence Diagram: Laporan Sistem Informasi

2.2.1.17 Inventarisasi Sistem Informasi



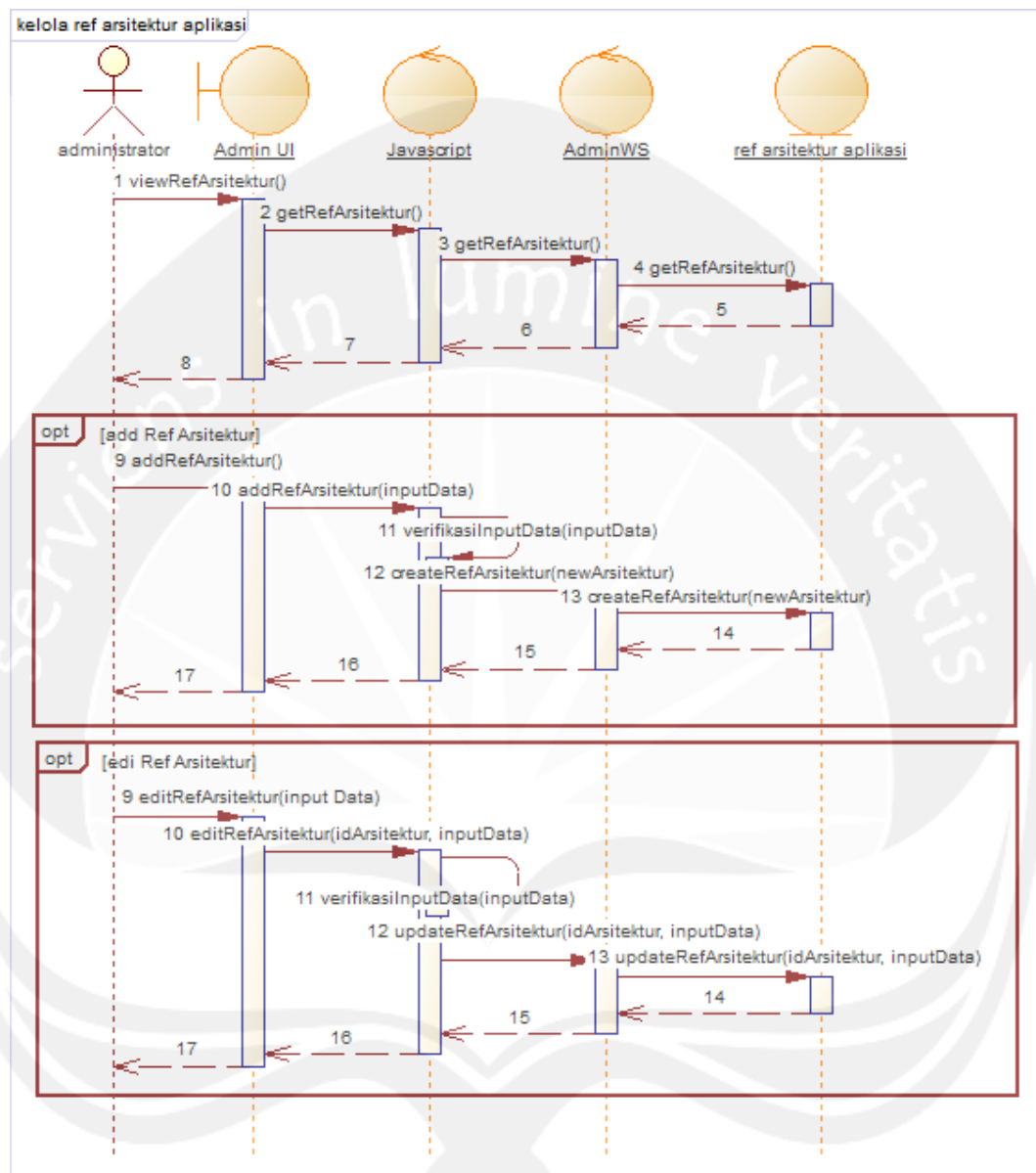
Gambar 19. Sequence Diagram: Inventarisasi Sistem Informasi

2.2.1.18 Kelola User



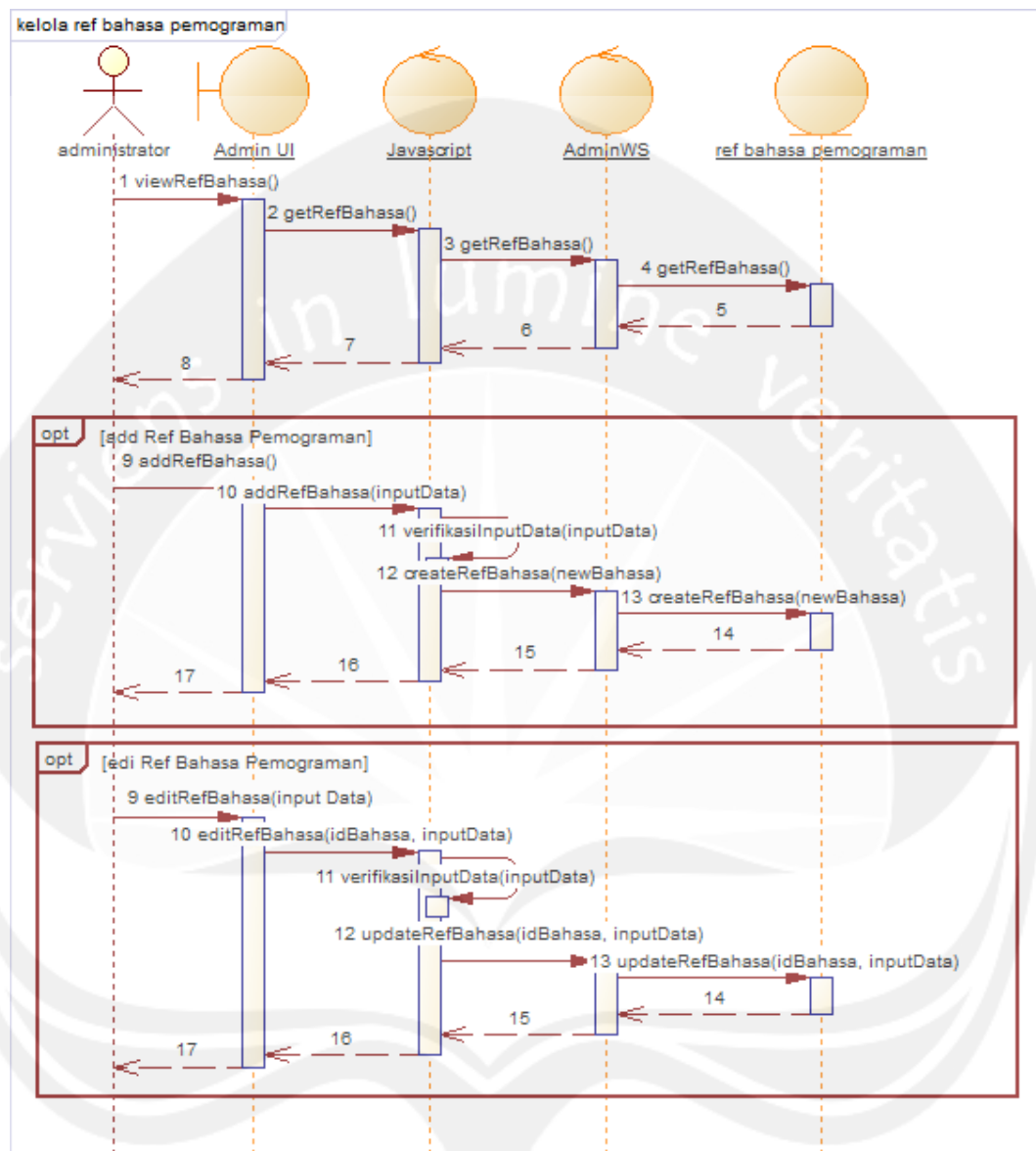
Gambar 20. Sequence Diagram: Kelola User

2.2.1.19 Kelola ref arsitektur aplikasi



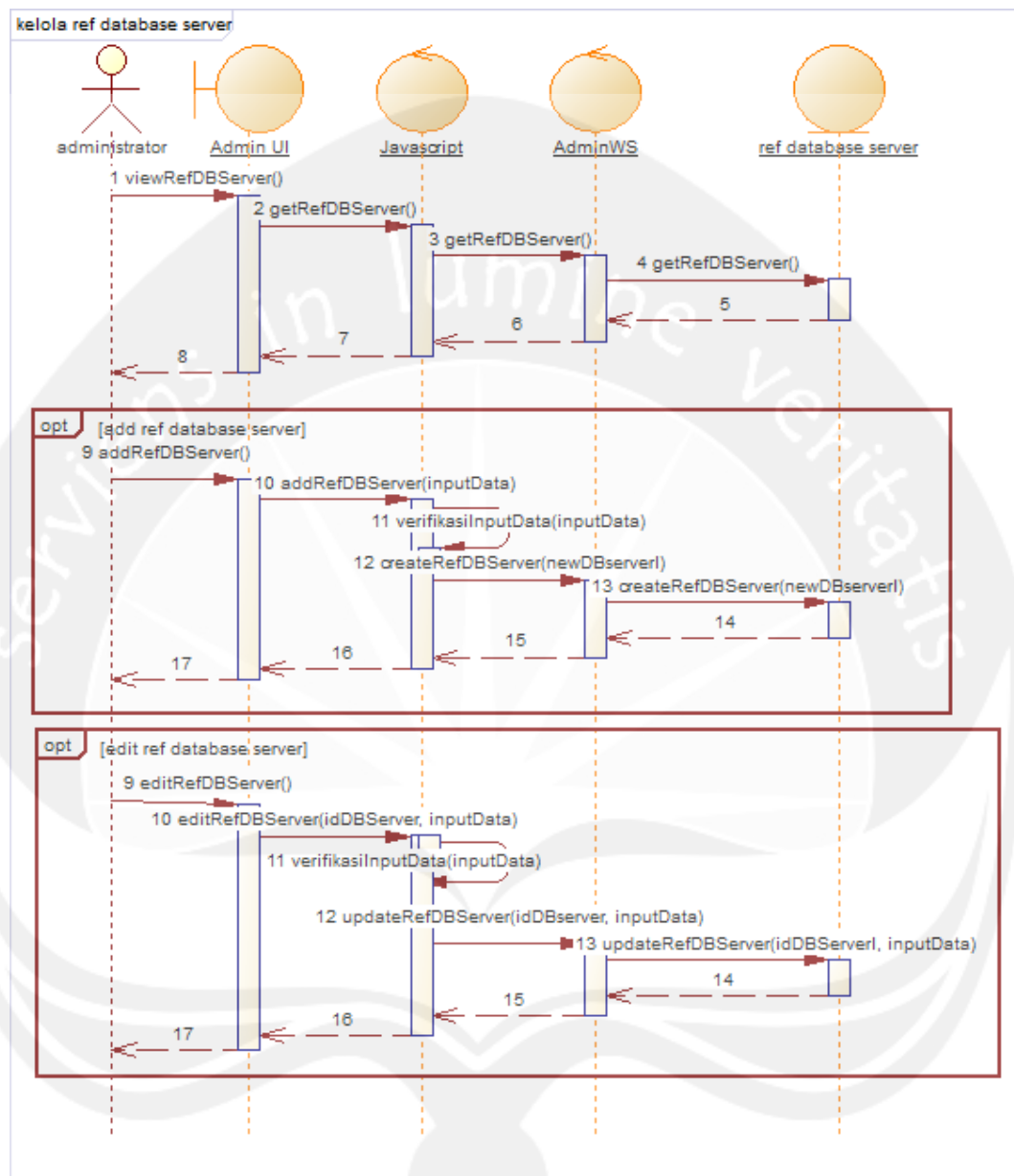
Gambar 21. Sequence Diagram: Kelola Ref Arsitektur

2.2.1.20 Kelola Ref Bahasa Pemrograman



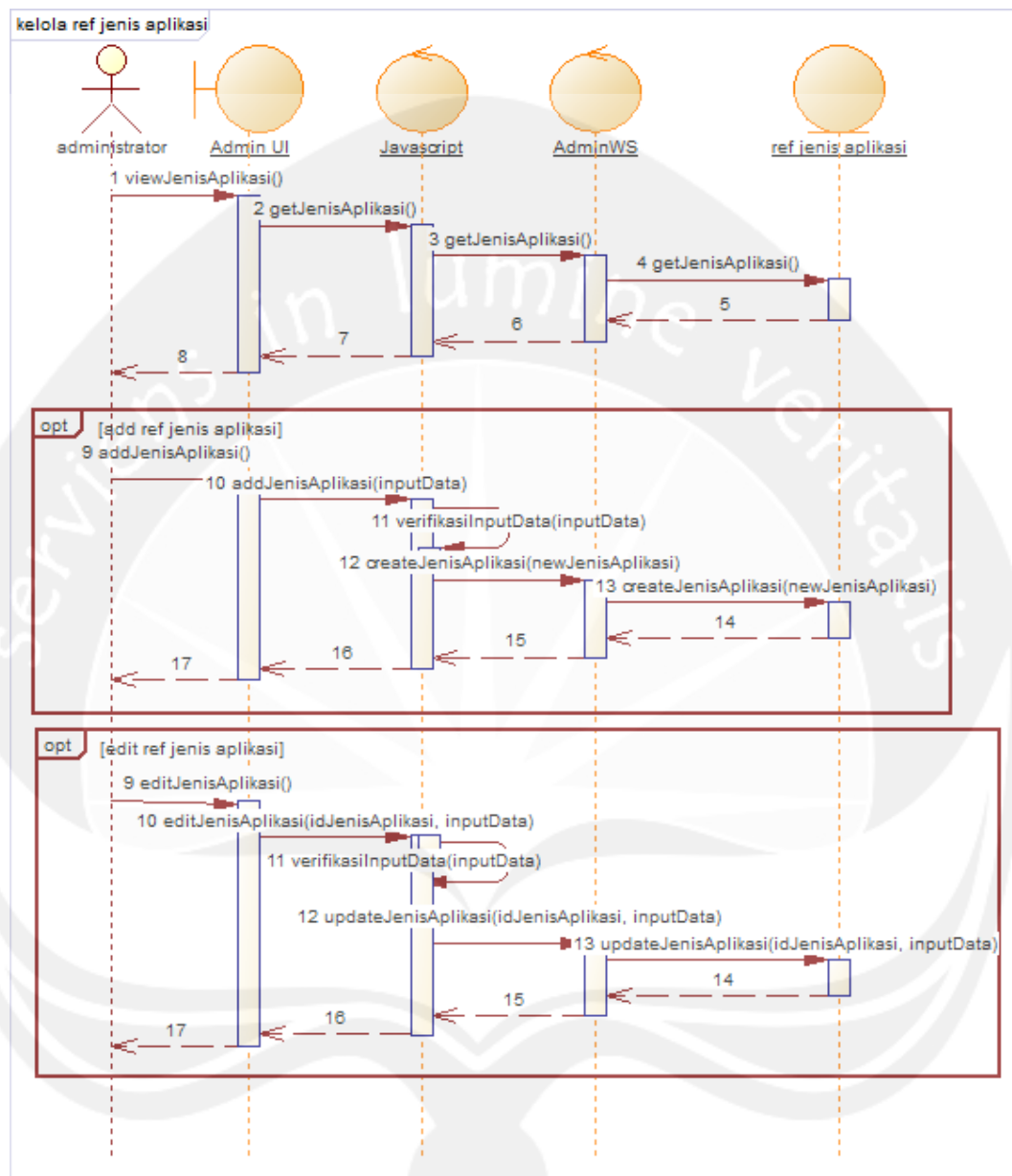
Gambar 22. Sequence Diagram: Kelola ref Bahasa Pemrograman

2.2.1.21 Kelola Ref Database Server



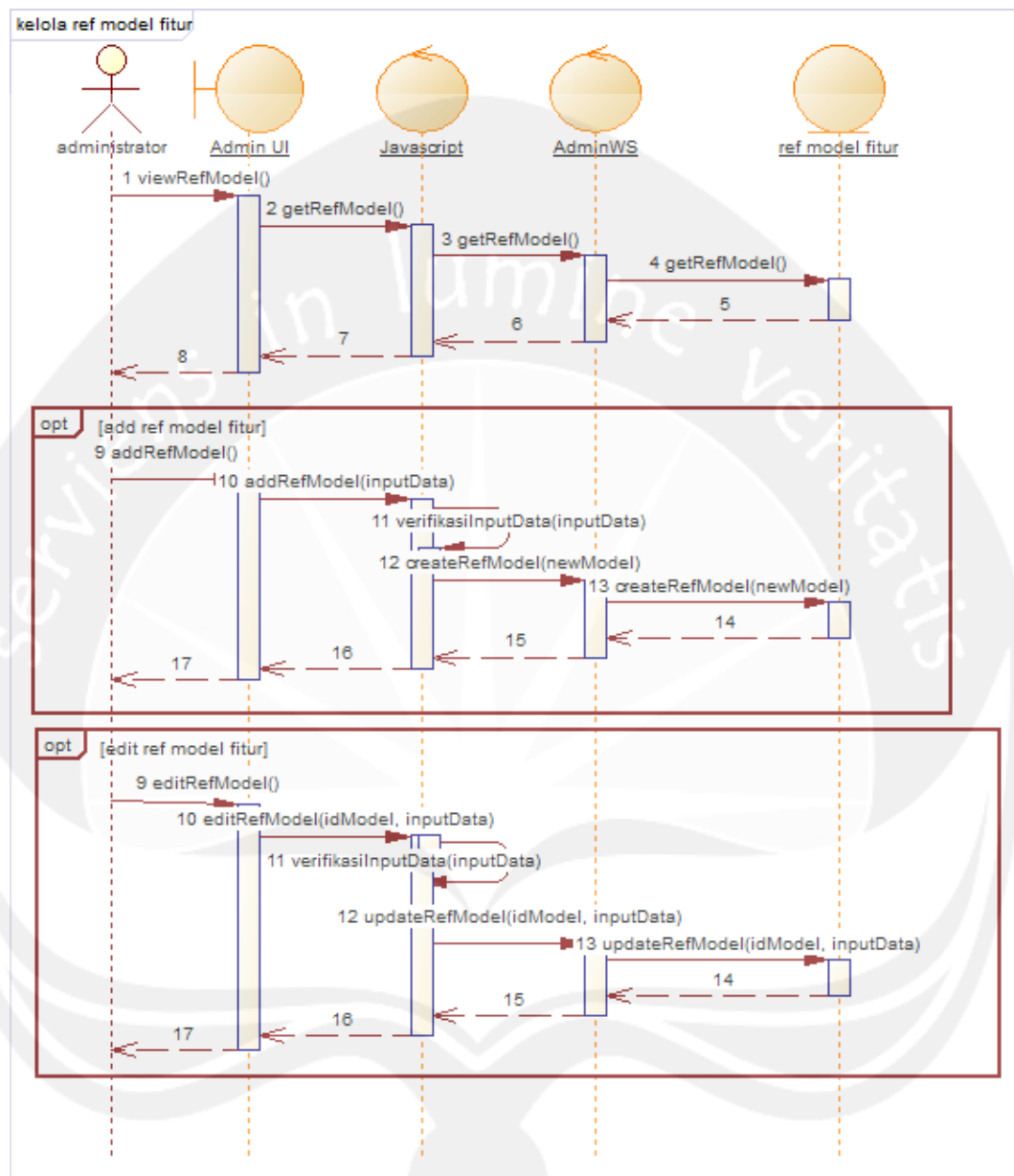
Gambar 23. Sequence Diagram: Kelola Ref Database Server

2.2.1.22 Kelola Jenis Aplikasi



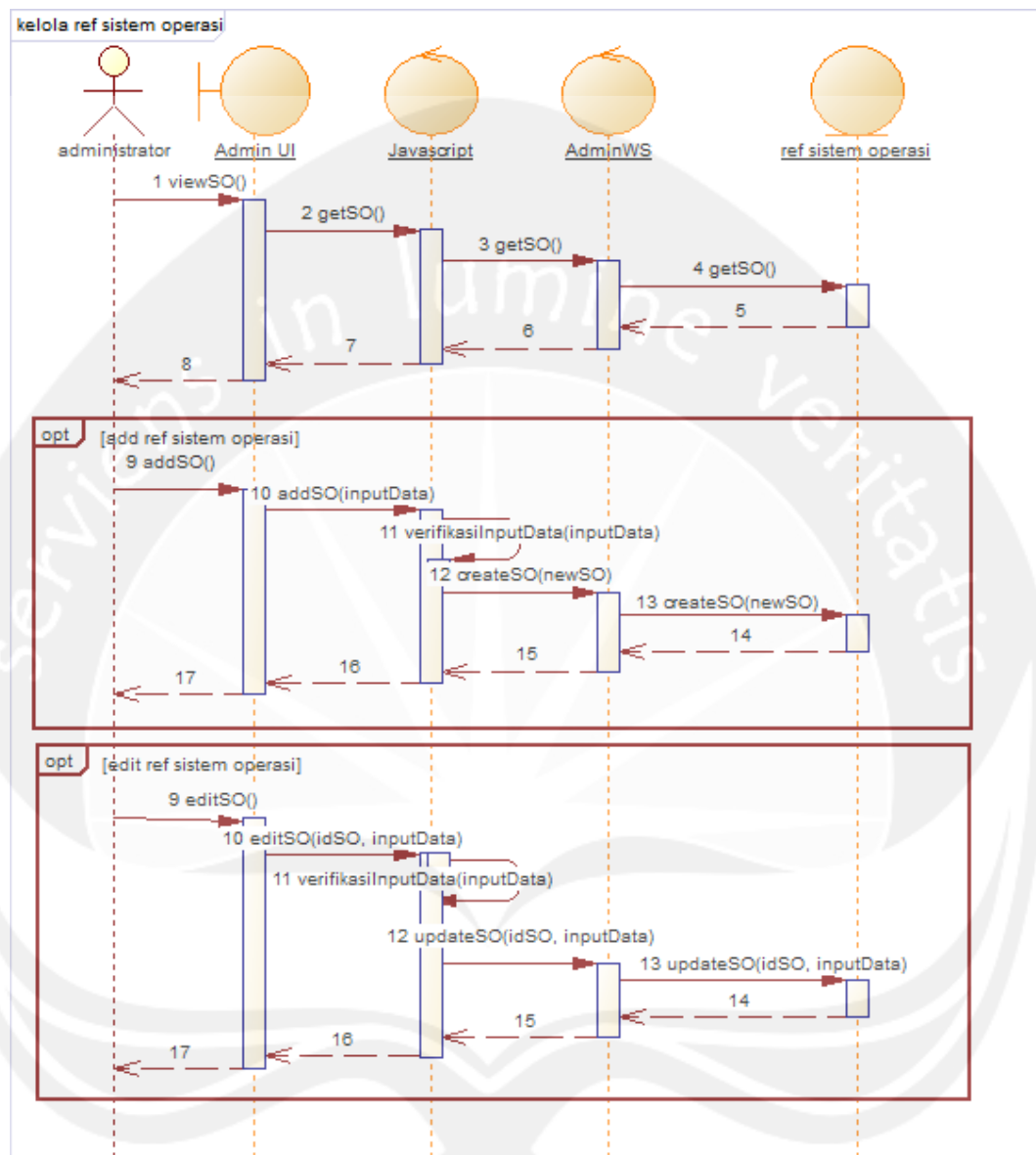
Gambar 24. Sequence Diagram: Kelola Ref Jenis Aplikasi

2.2.1.23 Kelola Ref Model Fitur



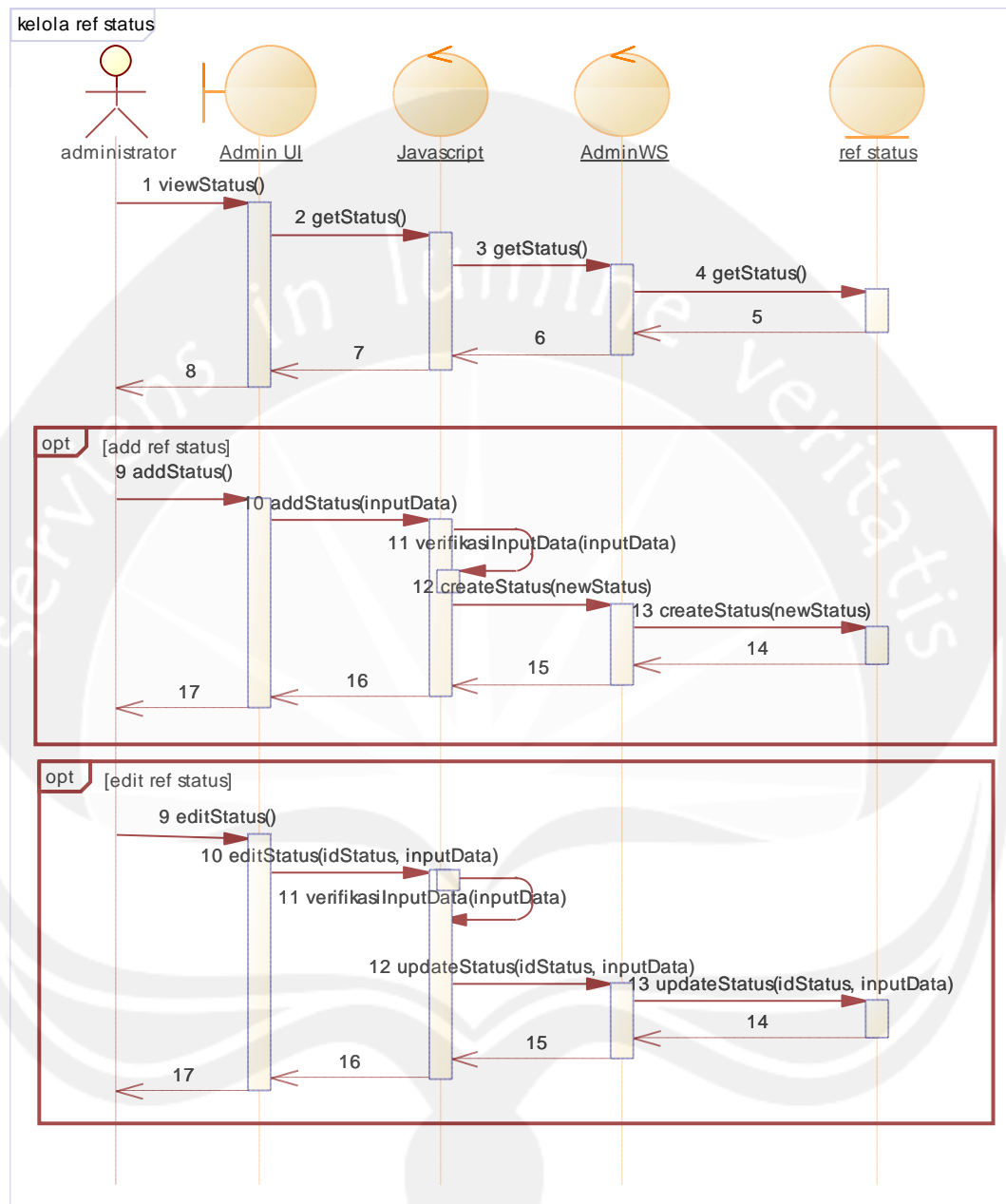
Gambar 25. Sequence Diagram: Kelola Ref Model Fitur

2.2.1.24 Kelola ref sistem operasi



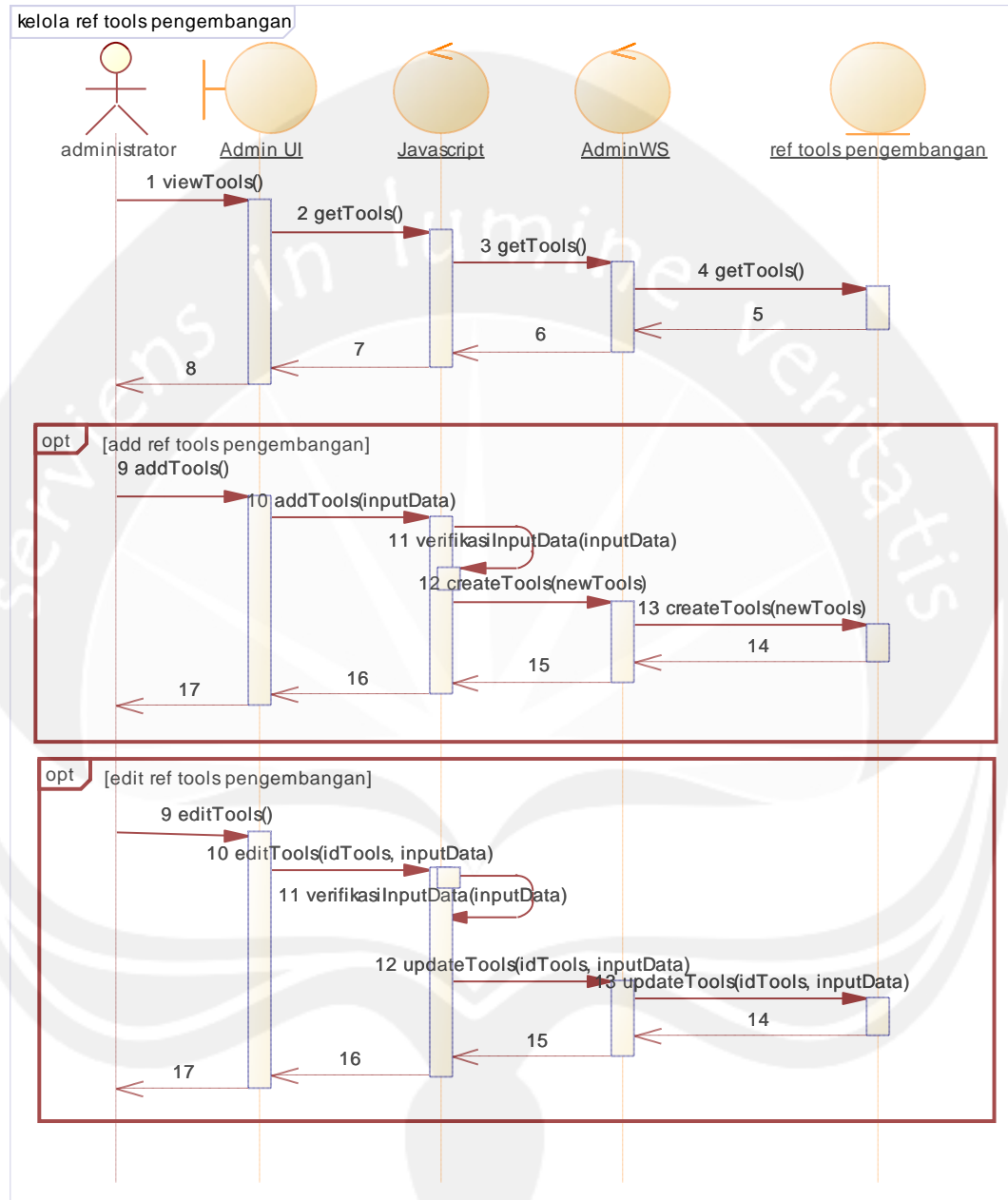
Gambar 26. Sequence Diagram: Kelola Ref Sistem Operasi

2.2.1.25 Kelola Ref Status



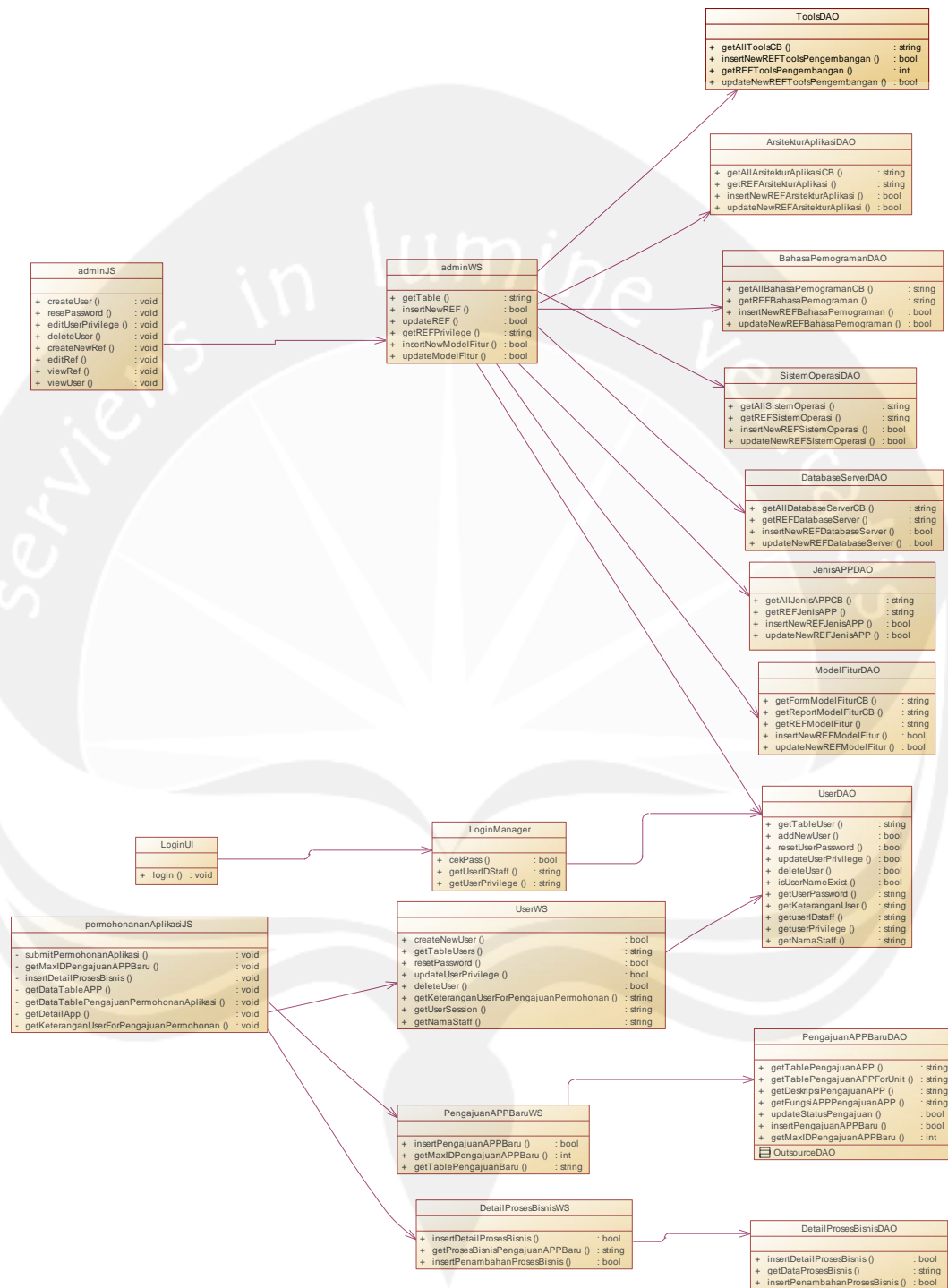
Gambar 27. Sequence Diagram: Kelola Ref Status

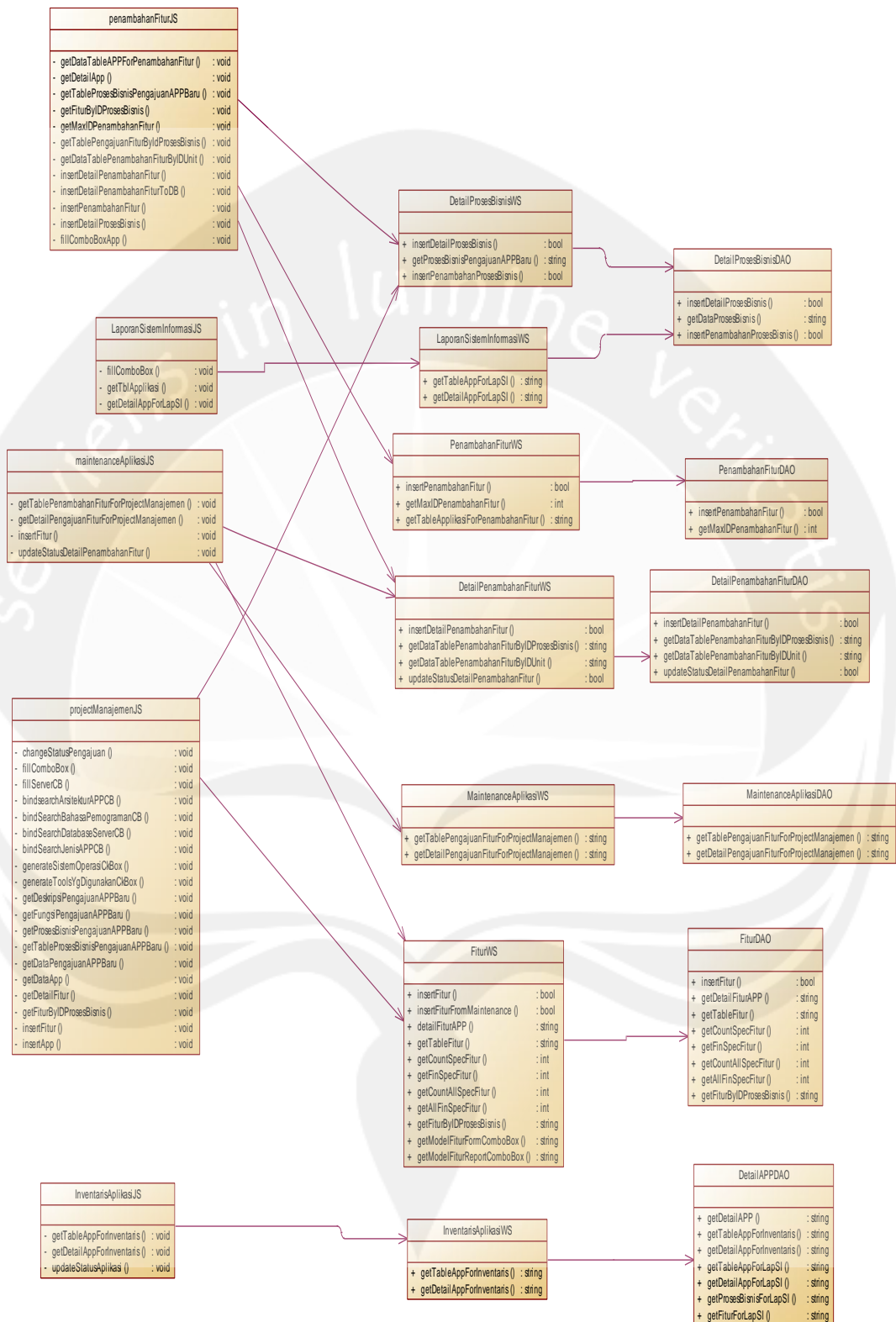
2.2.1.26 Kelola Ref Tools Pengembangan

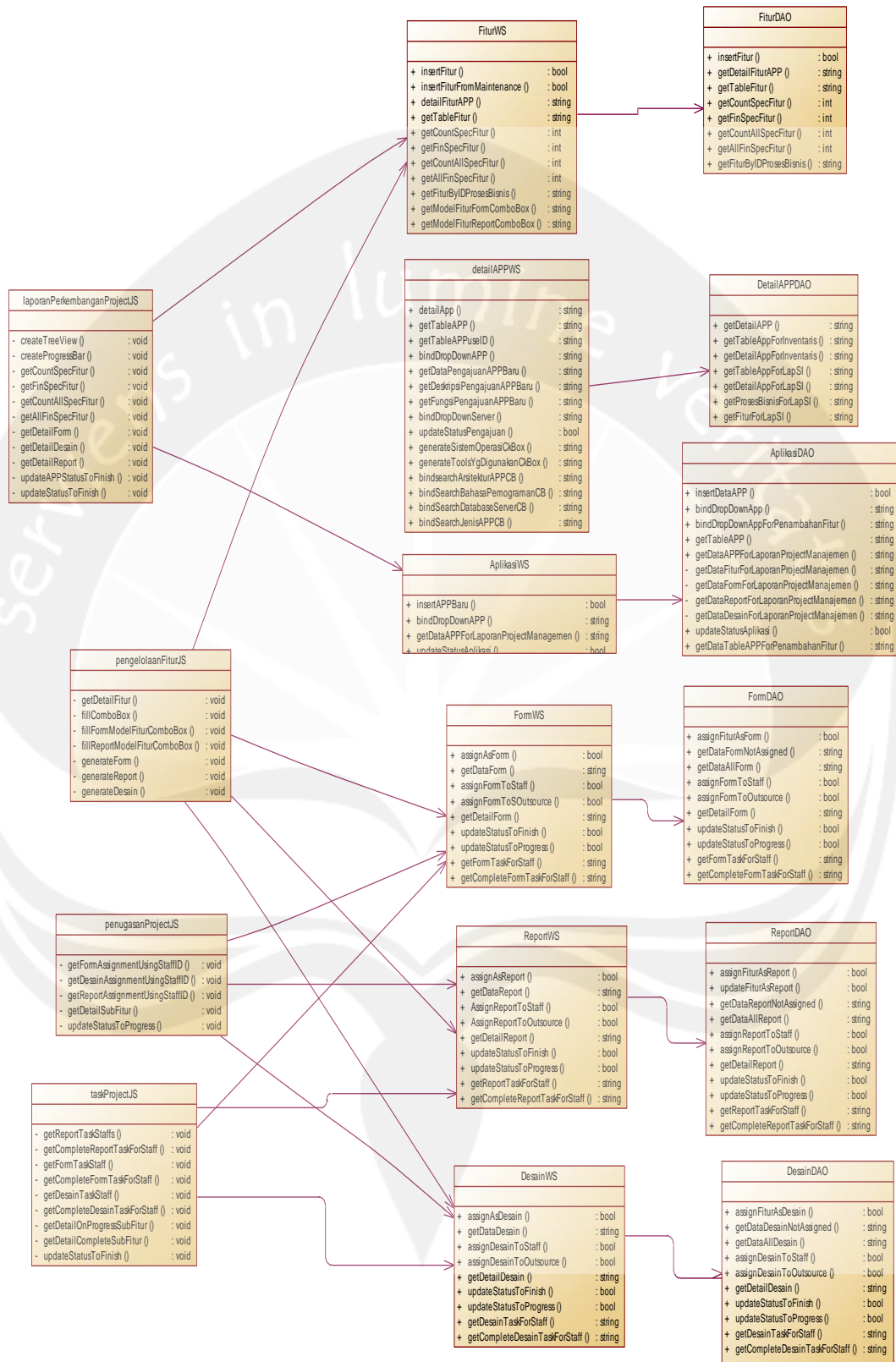


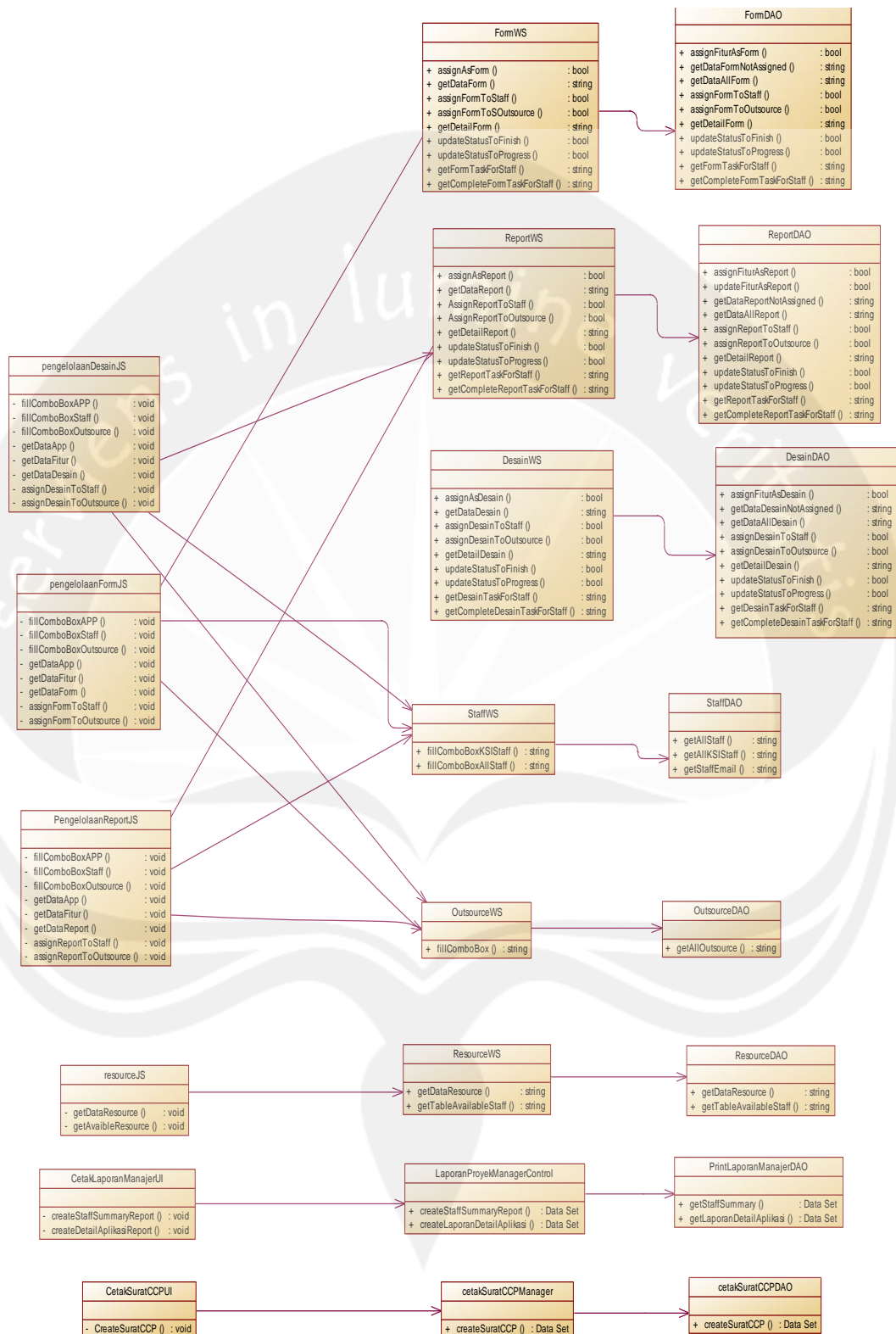
Gambar 28. Sequence Diagram: Kelola Ref Tools Pengembangan

2.2.2 Class Diagram









Gambar 29. Class Diagram

2.2.3 Class Diagram Specific Descriptions

2.2.3.1 Spesific Design Class adminJs

AdminJS	<<boundary>>
<p>- createUser(username : string, idStaff : int, privilege : string): void Operasi ini digunakan untuk membuat user baru.</p> <p>- resetPassword(username : string) : void Operasi ini digunakan untuk mengubah password user.</p> <p>- editUserPrivilege(username : string, privilege : string) : void Operasi ini digunakan untuk mengubah privilege user.</p> <p>- deleteUser(username : string) : void Operasi ini digunakan untuk menghapus data user.</p> <p>- createNewRef(namaRef : string, jenisRef : string) : void Operasi ini digunakan untuk membuat ref baru.</p> <p>- editRef(namaRef : string, jenisRef : string, IdRef : int): void Operasi ini digunakan untuk mengubah ref.</p> <p>- viewRef(jenisRef : string) : void Operasi ini digunakan untuk menampilkan data ref.</p> <p>- viewUser() : void Operasi ini digunakan untuk menampilkan data user.</p>	

2.2.3.2 Spesific Design Class LoginUI

LoginUI	<<boundary>>
<p>- login() : void Operasi ini digunakan untuk mengambil data login yang</p>	

diinputkan oleh user.

2.2.3.3 Spesific Design Class PermohonanJS

PermohonanJS	<<boundary>>
<p>- submitPermohonanAplikasi(newPermohonan : permohonan) : void Operasi ini digunakan untuk membuat sebuah permohonan pembuatan aplikasi baru.</p> <p>- getMaxIDpengajuanAPPBaru() : void Operasi ini digunakan untuk mengambil id pengajuan terbaru.</p> <p>- insertDetailProsesBisnis(newProsesBisnis : prosesBisnis) : void Operasi ini digunakan untuk menambahkan proses bisnis baru.</p> <p>- getDataTableAPP() : void Operasi ini digunakan untuk mengambil table aplikasi.</p> <p>- getDataTablePengajuanPermohonanAplikasi() : void Operasi ini digunakan untuk mengambil table pengajuan permohonan aplikasi.</p> <p>- getDetailApp(idApp : int) : void Operasi ini digunakan untuk mengambil detail aplikasi yang dipilih.</p> <p>- getKeteranganUserForPengajuanPermohonan(username : string) : void Operasi ini digunakan untuk mengambil keterangan user yang mengajukan permohonan.</p>	

2.2.3.4 Spesific Design Class penambahanFiturJS

PenambahanFiturJS		<<boundary>>
Program Studi Teknik Informatika	DPPL – SIKSI	39/ 100
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

<p>- getDataTableAPPForPenambahanFitur(): void</p> <p>Operasi ini digunakan untuk mengambil table aplikasi.</p> <p>- getDetailApp(idApp : int): void</p> <p>Operasi ini digunakan untuk mengambil detail aplikasi yang dipilih.</p> <p>- getTableProsesBisnisPengajuanAPPBaru (IdPengajuan : int) : void</p> <p>Operasi ini digunakan untuk mengambil table proses bisnis dari pengajuan aplikasi.</p> <p>- getFiturByIDProsesBisnis (idProsesBisnis : int) : void</p> <p>Operasi ini digunakan untuk mengambil table fitur berdasarkan id proses bisnis.</p> <p>- getMaxIDPenambahanFitur() : void</p> <p>Operasi ini digunakan untuk mengambil ID penambahan fitur.</p> <p>- getTablePengajuanFiturByIDProsesBisnis (idProsesBisnis : int): void</p> <p>Operasi ini digunakan untuk mengambil table pengajuan penambahan fitur berdasarkan id proses bisnis.</p> <p>- getDataTablePenambahanFiturByIDUnit(idUnit : int) : void</p> <p>Operasi ini digunakan untuk mengambil table penambahan fitur berdasarkan id unit user.</p> <p>- insertDetailPenambahanFitur() : void</p> <p>Operasi ini digunakan untuk memulai prosedur memasukkan data detail penambahan fitur.</p> <p>- insertDetailPenambahanFitur (newPenambahanFitur : penambahanFitur) : void</p> <p>Operasi ini digunakan untuk memassingkan data detail</p>	
Program Studi Teknik Informatika	DPPL – SIKSI
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika	

penambahan fitur ke control untuk dimasukkan kedalam database.

- insertPenambahanFitur() : void

Operasi ini digunakan untuk menambahkan penambahan fitur baru.

- insertDetailProsesBisnis() : void

Operasi ini digunakan untuk menambahkan detail proses bisnis baru.

- fillComboBoxAPP() : void

Operasi ini digunakan untuk mengisi data combo box.

2.2.3.5 Spesific Design Class LaporanSistemInformasiJS

LaporanSistemInformasiJS	<<boundary>>
- fillComboBoxAPP() : void Operasi ini digunakan untuk mengisi data combo box. - getTblAPP() : void Operasi ini digunakan untuk mengambil table aplikasi. - getDetailAppForLapSI(idApp : int): void Operasi ini digunakan untuk mengambil detail aplikasi yang dipilih.	

2.2.3.6 Spesific Design Class maintenancaAplikasiJS

maintenancaAplikasiJS	<<boundary>>
- getTablePenambahanFiturForProjectManager() : void Operasi ini digunakan untuk mengambil table penambahan fitur. - getDetailPengajuanFiturForProjectManajger	

(idPenambahanFitur : int): void

Operasi ini digunakan untuk mengambil detail penambahan fitur berdasarkan idpenambahan fitur.

- insertFitur(newFitur : fitur) : void

Operasi ini digunakan untuk membuat fitur baru.

- updateStatusDetailPenambahanFitur

(idDetailPenambahanFitur : int) : void

Operasi ini digunakan untuk mengubah status detail penambahan fitur.

2.2.3.7 Spesific Design Class projectManajemenJS

projectManajemenJS	<<boundary>>
- changeStatusPengajuan(idPengajuan : int): void Operasi ini digunakan untuk mengubah status pengajuan aplikasi baru.	
- getDeskripsiPengajuanAPPBaru(idPengajuan :int) : void Operasi ini digunakan untuk mengambil deskripsi dari pengajuan aplikasi.	
- getFungsiPengajuanAPPBaru(idPengajuan :int) : void Operasi ini digunakan untuk mengambil fungsi dari pengajuan aplikasi.	
- getProsesBisnisPengajuanAPPBaru(idPengajuan :int) : void Operasi ini digunakan untuk mengambil proses bisnis dari pengajuan aplikasi.	
- getTableProsesBisnisPengajuanAPPBaru(idPengajuan : int) void Operasi ini digunakan untuk mengambil proses bisnis dari pengajuan aplikasi	

- getDataPengajuanAPPBaru() : void

Operasi ini digunakan untuk mengambil data pengajuan aplikasi baru.

- getDetailFitur(idFitur : int) : void

Operasi ini digunakan untuk mengambil data detail fitur.

- getFiturByIDProsesBisnis(idProsesBisnis : int) : void

Operasi ini digunakan untuk mengambil data fitur berdasarkan id proses bisnis.

- insertApp(newAplikasi : aplikasi) : void

Operasi ini digunakan untuk membuat aplikasi.

- insertFitur(newFitur : fitur) : void

Operasi ini digunakan untuk membuat fitur baru.

2.2.3.8 Spesific Design Class InventarisAplikasiJS

InventarisAplikasiJS	<<boundary>>
- getTableAppForInventaris() : void Operasi ini digunakan untuk mengambil table aplikasi - - getDetailAppForInventaris(idApp : int) : void Operasi ini digunakan untuk mengambil detail aplikasi - - updateStatusAplikasi(idApp : int) : void Operasi ini digunakan untuk mengubah status Aplikasi.	

2.2.3.9 Spesific Design Class

laporanPerkembanganProjectJS

laporanPerkembanganProjectJS	<<boundary>>
<p>- createTreeView(): void Operasi ini digunakan untuk membuat tampilan treeview.</p> <p>- createProgressBar(username : string) : void Operasi ini digunakan untuk membuat progress bar.</p> <p>- getCountSpecFitur(idFitur : int) : void Operasi ini digunakan untuk menghitung jumlah fitur yang dimiliki satu fitur.</p> <p>- getFinSpecFitur(idFitur: int) : void Operasi ini digunakan untuk menghitung jumlah fitur yang sudah selesai dari satu fitur.</p> <p>- getCountAllSpecFitur(idAplikasi : int) : void Operasi ini digunakan untuk menghitung jumlah fitur yang dimiliki satu aplikasi.</p> <p>- getAllFinSpecFitur(idAplikasi : int): void Operasi ini digunakan untuk menghitung jumlah fitur yang sudah selesai dari satu fitur.</p> <p>- getDetailForm(idForm : int) : void Operasi ini digunakan untuk menampilkan detail form.</p> <p>- getDetailDesain (idDesain : int) : void Operasi ini digunakan untuk menampilkan detail desain.</p> <p>- getDetailReport(idReport : int) : void Operasi ini digunakan untuk menampilkan detail report.</p> <p>- UpdateAPPStatusToFinish(idAPP : int) void Operasi ini digunakan untuk mengubah status aplikasi.</p>	

2.2.3.10 Spesific Design Class pengelolaanFiturJS

pengelolaanFiturJS	<<boundary>>
- getDetailFitur(idFitur : int): void Operasi ini digunakan untuk mengambil detail fitur.	
- generateForm(newForm : form) : void Operasi ini digunakan untuk membuat form baru.	
- generateReport(newReport : report) : void Operasi ini digunakan untuk membuat report baru.	
- generateDesain(newDesain : desain) : void Operasi ini digunakan untuk membuat desain baru.	

2.2.3.11 Spesific Design Class penugasanProjectJS

penugasanProjectJS	<<boundary>>
- getFormAssignmentUsingStaffID(idStaff : int): void Operasi ini digunakan untuk mengambil tugas berupa form yang ditugaskan untuk staff.	
- getReportAssignmentUsingStaffID(idStaff : int): void Operasi ini digunakan untuk mengambil tugas berupa report yang ditugaskan untuk staff.	
- getDesainAssignmentUsingStaffID(idStaff : int): void Operasi ini digunakan untuk mengambil tugas berupa desain yang ditugaskan untuk staff.	
- getDetailSubFitur(jenis : string, id : int): void Operasi ini digunakan untuk mengambil detail dari subfitur desain yang ditugaskan untuk staff.	
- updateStatusToProgress(jenis : string, id: int): void Operasi ini digunakan untuk mengubah status form, desain, atau report yang ditugaskan untuk staff.	

2.2.3.12 Spesific Design Class taskProjectJS

taskProjectJS	<<boundary>>
<p>- getReportTaskStaffs(idStaff : int): void Operasi ini digunakan untuk mengambil report yang sedang dikerjakan oleh staff.</p> <p>- getCompleteReportTaskForStaff(idStaff : int) : void Operasi ini digunakan untuk mengambil report yang sudah selesai dikerjakan oleh staff.</p> <p>- getFormTaskStaffs(idStaff : int): void Operasi ini digunakan untuk mengambil form yang sedang dikerjakan oleh staff.</p> <p>- getCompleteFormTaskForStaff(idStaff : int) : void Operasi ini digunakan untuk mengambil form yang sudah selesai dikerjakan oleh staff.</p> <p>- getDesainTaskStaffs(idStaff : int): void Operasi ini digunakan untuk mengambil desain yang sedang dikerjakan oleh staff.</p> <p>- getCompleteDesainTaskForStaff(idStaff : int) : void Operasi ini digunakan untuk mengambil desain yang sudah selesai dikerjakan oleh staff.</p> <p>- getDetailOnProgressSubFitur(jenis : string, id : int) : void Operasi ini digunakan untuk mengambil detail subfitur yang sedang dikerjakan oleh staff.</p> <p>- getCompleteDesainTaskForStaff(jenis : string, id : int) : void Operasi ini digunakan untuk mengambil detail subfitur yang sudah selesai dikerjakan oleh staff.</p>	

2.2.3.13 Spesific Design Class pengelolaanDesainJS

pengelolaanDesainJS	<<boundary>>
<p>- getDataAPP() : void Operasi ini digunakan untuk mengambil data aplikasi.</p> <p>- getDataFitur(idApp : int) : void Operasi ini digunakan untuk mengambil data fitur.</p> <p>- getDataDesain(idFitur : int) : void Operasi ini digunakan untuk mengambil data desain.</p> <p>- assignDesainToStaff(idStaff : int) : void Operasi ini digunakan untuk menugaskan desain kepada staff.</p> <p>- assignDesainToOutsource(idOutsource: int) : void Operasi ini digunakan untuk menugaskan desain kepada outsource.</p>	

2.2.3.14 Spesfic Design Class pengelolaanFormJS

pengelolaanFormJS	<<boundary>>
<p>- getDataAPP() : void Operasi ini digunakan untuk mengambil data aplikasi.</p> <p>- getDataFitur(idApp : int) : void Operasi ini digunakan untuk mengambil data fitur.</p> <p>- getDataForm(idFitur : int) : void Operasi ini digunakan untuk mengambil data form.</p> <p>- assignFormToStaff(idStaff : int) : void Operasi ini digunakan untuk menugaskan form kepada staff.</p> <p>- assignFormToOutsource(idOutsource: int) : void</p>	

Operasi ini digunakan untuk menugaskan form kepada outsource.

2.2.3.15 Spesific Design Class pengelolaanReportJS

pengelolaanReportJS	<<boundary>>
<p>- getDataAPP() : void Operasi ini digunakan untuk mengambil data aplikasi.</p> <p>- getDataFitur(idApp : int) : void Operasi ini digunakan untuk mengambil data fitur.</p> <p>- getDataReport(idFitur : int) : void Operasi ini digunakan untuk mengambil data report.</p> <p>- assignReportToStaff(idStaff : int) : void Operasi ini digunakan untuk menugaskan report kepada staff.</p> <p>- assignReportToOutsource(idOutsource: int) : void Operasi ini digunakan untuk menugaskan report kepada outsource.</p>	

2.2.3.16 Spesific Design Class resourceJS

resourceJS	<<boundary>>
<p>- getDataResource(idStatus : int, idStaff: int): void Operasi ini digunakan untuk mengambil data staff yang sedang mengerjakan tugas.</p> <p>- getAvailableResource() : void Operasi ini digunakan untuk mengambil data staff yang tidak sedang mengerjakan tugas.</p>	

2.2.3.17 Spesific Design Class CetakLaporanManajerUI

Program Studi Teknik Informatika	DPPL – SIKSI	48/ 100
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

CetakLaporanManajerUI	<<boundary>>
- createStaffSummaryReport(tanggalSelesaiFrom : string, tanggalSelesaiTo: string) : void Operasi ini digunakan untuk membuat laporan staff summary. - createDetailAplikasiReport(tanggalSelesaiFrom : string, tanggalSelesaiTo: string) : void Operasi ini digunakan untuk membuat laporan detail aplikasi.	

2.2.3.18 Spesific Design Class CetakSuratCCPUI

CetakSuratCCPUI	<<boundary>>
- createSuratCCP(idStaff : int, tanggalSelesaiFrom : string, tanggalSelesaiTo : string): void Operasi ini digunakan untuk membuat membuat laporan surat CCP.	

2.2.3.19 Spesific Design Class adminWS

adminWS	<<control>>
+ getTable(jenis : string): strin Operasi ini digunakan untuk mengambil table pengelolaan dari model + insertNewREF(namaRef : string : jenisRef : string): bool Operasi ini digunakan untuk membuat ref baru sesuai dengan parameter jenis ref.	


```
+ updateRef(namaRef : string, jenisRef : string, IdRef : int): bool
```

Operasi ini digunakan untuk mengubah ref.

```
+ insertNewRefModelFitur(deskripsi : string, kompleksitas : string, kodeModel : string): bool
```

Operasi ini digunakan untuk menambah ref model fitur.

```
+ updateModelFitur(deskripsi : string, kompleksitas : string, kodeModel : string, idModelFitur : int): bool
```

Operasi ini digunakan untuk mengubah ref model fitur.

2.2.3.20 Spesific Design Class LoginManager

LoginManager	<<control>>
<pre>+ cekPass(username : string): bool</pre> <p>Operasi ini digunakan untuk memeriksa kecocokan password yang diinputkan dengan password user yang berada didatabase.</p> <pre>+ getUserIDStaff(username : string): string</pre> <p>Operasi ini digunakan untuk mengambil data id staff dari user yang login.</p> <pre>+ getUserPrivilege(username : string): string</pre> <p>Operasi ini digunakan untuk mengambil privilege dari user yang login.</p>	

2.2.3.21 Spesific Design Class UserWS

UserWS	<<control>>
<pre>+ createNewUser(newUser : user): bool</pre> <p>Operasi ini digunakan untuk membuat user baru dengan</p>	

menambahkannya kedalam database.

+ getTableUser(): strin

Operasi ini digunakan untuk mengambil table list user aplikasi.

+ resetPassword(username : string): bool

Operasi ini digunakan untuk mengubah password user menjadi default password.

+ updateUserPrivilege(username : string, privilege: string): bool

Operasi ini digunakan untuk mengubah privilege dari user.

+ deleteUser(username : string): bool

Operasi ini digunakan untuk menghapus user.

+ getKeteranganUserForPengajuanPermohonan(username : string): strin

Operasi ini digunakan untuk mengambil keterangan user yang mengajukan permohonan.

+ getUserSession() : string

Operasi ini digunakan untuk mengambil data user yang disimpan dalam session.

+getNamaStaff(username : string) : void

Operasi ini digunakan untuk mengambil nama user.

2.2.3.22 Spesific Design Class PengajuanAPPBaruWS

PengajuanAPPBaruWS	<<control>>
+ insertPengajuanAPPBBaru(NewPengajuanBaru : pengajuanBaru): void Operasi ini digunakan untuk memasukkan data pengajuan pembuatan aplikasi baru.	

Program Studi Teknik Informatika	DPPL – SIKSI	51/ 100
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

+ getMaxIDpengajuanAPPBaru() : int

Operasi ini digunakan untuk mengambil id pengajuan terbaru.

+ getTablePengajuanAPPBaru() : string

Operasi ini digunakan untuk mengambil table pengajuan permohonan aplikasi.

2.2.3.23 Spesific Design Class detailProsesBisnisWS

detailProsesBisnisWS	<<control>>
<p>+ insertDetailProsesBisnis(newProsesBisnis : prosesBisnis) : bool</p> <p>Operasi ini digunakan untuk menambahkan proses bisnis baru.</p> <p>+ getProsesBisnisPengajuanAPPBaru(IdPengajuan : int) : string</p> <p>Operasi ini digunakan untuk mengambil table proses bisnis dari pengajuan aplikasi.</p> <p>+ insertDetailProsesBisnis(newProsesBisnis : prosesBisnis) : bool</p> <p>Operasi ini digunakan untuk menambahkan detail proses bisnis baru.</p>	

2.2.3.24 Spesific Design Class LaporanSistemInformasiWS

LaporanSistemInformasiWS	<<control>>
<p>+ getTableAppForLapSI(idStatus : int): string</p> <p>Operasi ini digunakan untuk mengambil table aplikasi dengan status tertentu untuk laporan SI.</p>	

+ getDetailAPPForLapSI(IdApp : int): string

Operasi ini digunakan untuk mengambil detail aplikasi yang dipilih

2.2.3.25 Spesific Design Class PenambahanFiturWS

PenambahanFiturWS	<<control>>
+ insertPenambahanFitur() : bool Operasi ini digunakan untuk menambahkan penambahan fitur baru.	
+ getMaxIDPenambahanFitur() : int Operasi ini digunakan untuk mengambil ID penambahan fitur.	
+ getDataTableAPPForPenambahanFitur(): string Operasi ini digunakan untuk mengambil table aplikasi.	

2.2.3.26 Spesific Design Class DetailPenambahanFiturWS

DetailPenambahanFiturWS	<<control>>
+ insertDetailPenambahanFitur() : bool Operasi ini digunakan untuk memulai prosedur memasukkan data detail penambahan fitur.	
+ getDataTablePenambahanFiturByIDUnit(idUnit : int) : string Operasi ini digunakan untuk mengambil table penambahan fitur berdasarkan id unit user.	
+ getFiturByIDProsesBisnis(idProsesBisnis : int) : string Operasi ini digunakan untuk mengambil table fitur	

berdasarkan id proses bisnis.

+ updateStatusDetailPenambahanFitur

(idDetailPenambahanFitur : int) : bool

Operasi ini digunakan untuk mengubah status detail penambahan fitur.

2.2.3.27 Spesific Design Class MaintenanceAplikasiWS

MaintenanceAplikasiWS	<<control>>
+ getTablePenambahanFiturForProjectManager() : string Operasi ini digunakan untuk mengambil table penambahan fitur.	
+ getDetailPengajuanFiturForProjectManajger (idPenambahanFitur : int): string Operasi ini digunakan untuk mengambil detail penambahan fitur berdasarkan idpenambahan fitur.	

2.2.3.28 Spesific Design Class InventarisAplikasiWS

InventarisAplikasiWS	<<control>>
+ getTableAppForInventaris(): string Operasi ini digunakan untuk mengambil table aplikasi +	
+ getDetailAppForInventaris(idApp : int) : string Operasi ini digunakan untuk mengambil detail aplikasi	

2.2.3.29 Spesific Design Class FiturWS

FiturWS	<<control>>
+ insertFitur(newFitur : fitur) : bool	

Operasi ini digunakan untuk membuat fitur baru.

+ insertFiturFromMaintenance(newFitur : fitur) : bool

Operasi ini digunakan untuk membuat fitur baru dari prosedur maintenance aplikasi.

+ detailFiturAPP() : string

Operasi ini digunakan untuk mengambil data detail fitur.

+ getTableFitur() : string

Operasi ini digunakan untuk mengambil table fitur.

+ getCountSpecFitur(idFitur : int) : int

Operasi ini digunakan untuk menghitung jumlah fitur yang dimiliki satu fitur.

+ getFinSpecFitur(idFitur: int) : int

Operasi ini digunakan untuk menghitung jumlah fitur yang sudah selesai dari satu fitur.

+ getCountAllSpecFitur(idAplikasi : int) : int

Operasi ini digunakan untuk menghitung jumlah fitur yang dimiliki satu aplikasi.

+ getAllFinSpecFitur(idAplikasi : int): int

Operasi ini digunakan untuk menghitung jumlah fitur yang sudah selesai dari satu fitur.

+ getFiturByIDProsesBisnis(idProsesBisnis : int) : string

Operasi ini digunakan untuk mengambil table fitur berdasarkan id proses bisnis.

2.2.3.30 Spesific Design Class detailAPPWS

detailAPPWS	<<control>>
+ detailAPP(idAPP : int): string	

Program Studi Teknik Informatika	DPPL – SIKSI	55/ 100
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Operasi ini digunakan untuk mengambil detail aplikasi.

+ getTableAPPUseID(idAPP : int) : string

Operasi ini digunakan untuk mengambil table aplikasi dengan kriteria Idapp.

+ getDataTablePengajuanAppBaru(namaAPP : string) : string

Operasi ini digunakan untuk mengambil table pengajuan permohonan aplikasi.

+ getDeskripsiPengajuanAPPBaru(idPengajuan :int) : string

Operasi ini digunakan untuk mengambil deskripsi dari pengajuan aplikasi.

+ getFungsiPengajuanAPPBaru(idPengajuan :int) : string

Operasi ini digunakan untuk mengambil fungsi dari pengajuan aplikasi.

+ getProsesBisnisPengajuanAPPBaru(idPengajuan :int) : string

Operasi ini digunakan untuk mengambil proses bisnis dari pengajuan aplikasi.

+ updateStatusPengajuan(idPengajuan : int): bool

Operasi ini digunakan untuk mengubah status pengajuan aplikasi baru.

2.2.3.31 Spesific Design Class AplikasiWS

AplikasiWS	<<control>>
+ insertApp(newAplikasi : aplikasi) : bool Operasi ini digunakan untuk membuat aplikasi. + updateStatusAplikasi(idApp : int, idStatus: id) : bool	

Operasi ini digunakan untuk mengubah status Aplikasi.

+ getDataAPPForLaporanProjectManajemen(): string

Operasi ini digunakan untuk mengambil data aplikasi yang sedang dikerjakan.

2.2.3.32 Spesific Design Class FormWS

FormWS	<<control>>
+ assignAsForm(newForm : form) : bool Operasi ini digunakan untuk membuat form baru.	
+ getDataForm(idFitur : int) : string Operasi ini digunakan untuk mengambil data form.	
+ assignFormToStaff(idStaff : int) : bool Operasi ini digunakan untuk menugaskan form kepada staff.	
+ assignFormToOutsource(idOutsource: int) : bool Operasi ini digunakan untuk menugaskan form kepada outsource.	
+ getFormTaskForStaff(idStaff : int): string Operasi ini digunakan untuk mengambil tugas berupa form yang ditugaskan untuk staff.	
+ getCompleteFormTaskForStaff(idStaff : int) : string Operasi ini digunakan untuk mengambil form yang sudah selesai dikerjakan oleh staff.	
+ getDetailForm(idForm : int) : string Operasi ini digunakan untuk mengambil detail form	
+ updateStatusToFinish(idForm : int, idStatus : int, updateform : form) : bool Operasi ini digunakan untuk mengubah status form menjadi finish.	

+ updateStatusToProgress(idForm : int, idStatus : int, updateform : form) : bool

Operasi ini digunakan untuk mengubah status form menjadi progress.

2.2.3.33 Spesific Design Class ReportWS

ReportWS	<<control>>
<p>+ assignAsReport(newReport : report) : bool Operasi ini digunakan untuk membuat report baru.</p> <p>+ getDataReport(idFitur : int) : string Operasi ini digunakan untuk mengambil data report.</p> <p>+ assignReportToStaff(idStaff : int) : bool Operasi ini digunakan untuk menugaskan report kepada staff.</p> <p>+ assignReportToOutsource(idOutsource: int) : bool Operasi ini digunakan untuk menugaskan report kepada outsource.</p> <p>+ getReportTaskForStaff(idStaff : int): string Operasi ini digunakan untuk mengambil tugas berupa report yang ditugaskan untuk staff.</p> <p>+ getCompleteReportTaskForStaff(idStaff : int) : string Operasi ini digunakan untuk mengambil report yang sudah selesai dikerjakan oleh staff.</p> <p>+ getDetailReport(idReport : int) : string Operasi ini digunakan untuk mengambil detail report</p> <p>+ updateStatusToFinish(idReport : int, idStatus : int, updatereport : report) : bool Operasi ini digunakan untuk mengubah status report menjadi finish.</p>	

+ updateStatusToProgress(idReport : int, idStatus : int, updatereport : report) : bool

Operasi ini digunakan untuk mengubah status report menjadi progress.

2.2.3.34 Spesific Design Class DesainWS

DesainWS	<<control>>
+ assignAsDesain(newDesain : desain) : bool Operasi ini digunakan untuk membuat desain baru.	
+ getDataDesain(idFitur : int) : string Operasi ini digunakan untuk mengambil data desain.	
+ assignDesainToStaff(idStaff : int) : bool Operasi ini digunakan untuk menugaskan desain kepada staff.	
+ assignDesainToOutsource(idOutsource: int) : bool Operasi ini digunakan untuk menugaskan desain kepada outsource.	
+ getDesainTaskForStaff(idStaff : int): string Operasi ini digunakan untuk mengambil tugas berupa desain yang ditugaskan untuk staff.	
+ getCompleteDesainTaskForStaff(idStaff : int) : string Operasi ini digunakan untuk mengambil desain yang sudah selesai dikerjakan oleh staff.	
+ getDetailDesain(idDesain : int) : string Operasi ini digunakan untuk mengambil detail desain	
+ updateStatusToFinish(idDesain : int, idStatus : int, updatedesain : desain) : bool Operasi ini digunakan untuk mengubah status desain menjadi finish.	

+ updateStatusToProgress(idDesain : int, idStatus : int, updatedesain : desain) : bool

Operasi ini digunakan untuk mengubah status desain menjadi progress.

2.2.3.35 Spesific Design Class StaffWS

StaffWS	<<control>>
+ fillComboBoxKSIStaff(IdUnit : int) : string Operasi ini digunakan untuk mengisi combobox staff KSI. + fillComboBoxStaff(): string Operasi ini digunakan untuk mengisi combobox staff.	

2.2.3.36 Spesific Design Class OutsourceWS

OutsourceWS	<<control>>
+ fillComboBox(): string Operasi ini digunakan untuk mengisi combobox outsource.	

2.2.3.37 Spesific Design Class ResourceWS

ResourceWS	<<control>>
+ getDataResource(IdStatus : int, IdStaf : int) : string Operasi ini digunakan untuk mengambil data resource yang sedang mengerjakan task. + getTableAvailableStaff () : string Operasi ini digunakan untuk staff yang tersedia.	

2.2.3.38 Spesific Design Class

LaporanProyekManajerControl

LaporanProyekManajerControl	<<control>>
+ createStaffSummaryReport(tanggalSelesaiFrom : string,tanggalSelesaiTo : string) : Data Set Operasi ini digunakan untuk membuat report Staff summary.	
+ createLaporanDetailAplikasi(tanggalSelesaiFrom : string,tanggalSelesaiTo : string)	
+ getDataDesain(idFitur : int) : Data Set Operasi ini digunakan untuk membuat report detail aplikasi.	

2.2.3.39 Spesific Design Class cetakSuratCCPManager

DesainWS	<<control>>
+ createSuratCCP(IdStaff:int, tanggalSelesaiFrom : string,tanggalSelesaiTo : string) : Data set Operasi ini digunakan untuk membuat surat CCP.	

2.2.3.40 Spesific Design Class ToolsDAO

ToolsDAO	<<entity>>
+ getAllTools() : String Operasi ini digunakan untuk mengambil semua data tools pengembangan dari database.	
+ insertREFToolsPengembangan(deskripsi : string): bool. Operasi ini digunakan untuk memasukkan data ref tools	

pengembangan baru ke dalam database.

+ updateNewRefToolsPengembangan(id : int, deskripsi : string): bool

Operasi ini digunakan untuk mengubah data tools pengembangan didatabase.

2.2.3.41 Spesific Design Class ArsitekturAplikasiDAO

ArsitekturAplikasiDAO	<<entity>>
+ getAllArsitekturAplikasi(): String Operasi ini digunakan untuk mengambil semua data arsitektur aplikasi dari database. + insertREFArsitekturAplikasi(deskripsi : string): bool. Operasi ini digunakan untuk memasukkan data ref arsitektur aplikasi baru ke dalam database. + updateNewRefArsitekturAplikasi(id : int, deskripsi : string): bool Operasi ini digunakan untuk mengubah data arsitektur aplikasi didatabase.	

2.2.3.42 Spesific Design Class BahasaPemrogramanDAO

BahasaPemrogramanDAO	<<entity>>
+ getAllBahasaPemrograman(): String Operasi ini digunakan untuk mengambil semua data bahasa pemrograman dari database. + insertREFBahasaPemrograman (deskripsi : string): bool.	

Operasi ini digunakan untuk memasukkan data ref bahasa pemrograman baru ke dalam database.

+ updateNewRefBahasaPemrograman (id : int, deskripsi : string): bool

Operasi ini digunakan untuk mengubah data bahasa pemrograman didatabase.

2.2.3.43 Spesific Design Class SistemOperasiDAO

SistemOperasiDAO	<<entity>>
+ getAllSistemOperasi(): String Operasi ini digunakan untuk mengambil semua data sistem operasi dari database. + insertREFSistemOperasi (deskripsi : string): bool. Operasi ini digunakan untuk memasukkan data ref sistem operasi baru ke dalam database. + updateNewRefSistemOperasi (id : int, deskripsi : string): bool Operasi ini digunakan untuk mengubah data sistem operasi didatabase.	

2.2.3.44 Spesific Design Class DatabaseServerDAO

DatabaseServerDAO	<<entity>>
+ getAllTools(): String Operasi ini digunakan untuk mengambil semua data database server dari database. + insertREFDatabaseServer (deskripsi : string): bool. Operasi ini digunakan untuk memasukkan data ref	

database server baru ke dalam database.

+ updateNewRefDatabaseServer(id : int, deskripsi : string): bool

Operasi ini digunakan untuk mengubah database server didatabase.

2.2.3.45 Spesific Design Class JenisAPPDAO

JenisAPPDAO	<<entity>>
+ getAllJenisAPP(): String Operasi ini digunakan untuk mengambil semua data jenis aplikasi pengembangan dari database. + insertREFJenisAPP (deskripsi : string): bool. Operasi ini digunakan untuk memasukkan data ref jenis aplikasi pengembangan baru ke dalam database. + updateNewRefJenisAPPP (id : int, deskripsi : string): bool Operasi ini digunakan untuk mengubah data jenis aplikasi didatabase.	

2.2.3.46 Spesific Design Class ModelFiturDAO

ModelFiturDAO	<<entity>>
+ getAllModelFitur(): String Operasi ini digunakan untuk mengambil semua data model fitur pengembangan dari database. + insertREFModelFitur (deskripsi : string): bool. Operasi ini digunakan untuk memasukkan data ref model fitur pengembangan baru ke dalam database.	

+ updateNewRefModelFiturP (id : int, deskripsi : string): bool

Operasi ini digunakan untuk mengubah data model fitur didatabase.

2.2.3.47 Spesific Design Class ModelFiturDAO

ModelFiturDAO	<<entity>>
+ getAllModelFitur(): String Operasi ini digunakan untuk mengambil semua data model fitur pengembangan dari database.	
+ insertREFModelFitur (deskripsi : string, kompleksitas :string, kodeModel : string): bool. Operasi ini digunakan untuk memasukkan data ref model fitur pengembangan baru ke dalam database.	
+ updateNewRefModelFiturP (id : int, deskripsi : string, kompleksitas :string, kodeModel : string): bool Operasi ini digunakan untuk mengubah data model fitur didatabase.	

2.2.3.48 Spesific Design Class UserDao

UserDAO	<<entity>>
+ getTableUser (): String Operasi ini digunakan untuk mengambil semua data user dari database.	
+ addNewUser (newUser : user): bool. Operasi ini digunakan untuk memasukkan data user baru ke dalam database.	

+ deleteUser (username : string): bool

Operasi ini digunakan untuk menghapus data user dengan mengubah nilai flag pada table user dalam database.

+ isUserNameExist (username : string): bool

Operasi ini digunakan untuk melakukan memeriksa apakah username sudah digunakan dalam database.

+ getUserPassword (username : string): string

Operasi ini digunakan untuk mengambil data password user dalam database.

+ getKeteranganUserForPengajuanPermohonan (username : string): string

Operasi ini digunakan untuk mengambil data keterangan user dalam database.

+ getUserIDstaff (username : string): string

Operasi ini digunakan untuk mengambil data idStaff user dalam database.

+ getUserPrivilege (username : string): string

Operasi ini digunakan untuk mengambil data privilege user dalam database.

+ getNamaStaff (idStaff : int): string

Operasi ini digunakan untuk mengambil data nama staff dalam database.

2.2.3.49 Spesific Design Class PengajuanAPPBaruDAO

PengajuanAPPBaruDAO	<<entity>>
+ getTablePengajuanAPP (string _namaAPP): String Operasi ini digunakan untuk mengambil semua data pengajuan aplikasi dengan kriteria nama aplikasi seperti pada parameter dari database.	

+ getTablePengajuanAPPForUnit (idUnit : int): string

Operasi ini digunakan untuk mengambil semua data pengajuan aplikasi yang diajukan oleh unit dari database.

+ getDeskripsiPengajuanAPP (idAPP : int): string

Operasi ini digunakan untuk mengambil data deskripsi pengajuan aplikasi tertentu dari database.

+ getFungsiAPPPengajuanAPP (idAPP : int): string

Operasi ini digunakan untuk mengambil data fungsi pengajuan aplikasi tertentu dari database.

+ updateStatusPengajuan (updatePengajuan : pengajuan): bool

Operasi ini digunakan untuk mengubah data pengajuan pada database.

+ insertPengajuanAPPBaru (updatePengajuan : pengajuan): bool

Operasi ini digunakan untuk memasukkan data pengajuan baru kedalam database.

+ getMaxIDPengajuanAPPBaru (updatePengajuan : pengajuan): int

Operasi ini digunakan untuk mengambil data idpengajuan terbaru dari database.

2.2.3.50 Spesific Design Class PenambahanFiturDAO

PenambahanFiturDAO	<<entity>>
+ insertPenambahanFitur (newPenambahanFitur : penambahanFitur): bool Operasi ini digunakan untuk memasukkan data penambahan fitur baru kedalam database.	

Program Studi Teknik Informatika	DPPL – SIKSI	67/ 100
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

+ getMaxIDPenambahanFitur () : int

Operasi ini digunakan untuk mengambil data id penambahan fitur terbaru dalam database.

2.2.3.51 Spesific Design Class DetailPenambahanFiturDAO

DetailPenambahanFiturDAO	<<entity>>
<p>+ insertDetailPenambahanFitur () : bool</p> <p>Operasi ini digunakan untuk memasukkan data detail penambahan fitur baru ke dalam database.</p> <p>+ getDataTablePenambahanFiturByIDProsesBisnis (idProsesBisnis : int) : string</p> <p>Operasi ini digunakan untuk mengambil data table penambahan fitur dari database dengan kriteria idProsesBisnis.</p> <p>+ getDataTablePenambahanFiturByIDUnit (idUnit : int) : bool</p> <p>Operasi ini digunakan untuk mengambil data table penambahan fitur dari database dengan kriteria idUnit.</p> <p>+ updateStatusDetailPenambahanFitur (idDetailpenambahanFitur : int) : bool</p> <p>Operasi ini digunakan untuk mengubah data status detail penambahan fitur didatabase.</p>	

2.2.3.52 Spesific Design Class MaintenanceAplikasiDAO

MaintenanceAplikasiDAO	<<entity>>
<p>+ getTablePengajuanFiturForProjectManajemen () : String</p> <p>Operasi ini digunakan untuk mengambil semua data</p>	

pengajuan penambahan fitur dari database.

**+ getDetailPengajuanFiturForProjectManajemen
(idDetailPenambahanFitur):String.**

Operasi ini digunakan untuk memasukkan data detail penambahan fitur baru ke dalam database.

2.2.3.53 Spesific Design Class FiturDAO

FiturDAO	<<entity>>
+ insertFitur (newFitur Fitur): bool Operasi ini digunakan untuk memasukkan data fitur baru ke dalam database.	
+ getDetailFiturAPP (idApp : int): string Operasi ini digunakan untuk mengambil data detail fitur aplikasi.	
+ getTableFitur (idApp : int): string Operasi ini digunakan untuk mengambil data table fitur dari database sesuai dengan idApp.	
+ getCountSpecFitur (idFitur : int): int Operasi ini digunakan untuk menghitung spesifikasi subfitur dalam database sesuai dengan idFitur.	
+ getFinSpecFitur (idFitur : int): int Operasi ini digunakan untuk menghitung spesifikasi subfitur yang sudah selesai dalam database sesuai dengan idFitur.	
+ getCountAllSpecFitur (IdApp : int): int Operasi ini digunakan untuk menghitung spesifikasi fitur dalam database sesuai dengan idApp.	
+ getAllFinSpecFitur (): int Operasi ini digunakan untuk menghitung spesifikasi	

fitur yang sudah selesai dalam database sesuai dengan idApp.

+ getFiturByIDProsesBisnis (): string

Operasi ini digunakan untuk mengambil data table fitur dari database sesuai dengan idProsesBisnis.

2.2.3.54 Spesific Design Class detailAppDAO

detailAppDAO	<<entity>>
<p>+ getDetailAPP (idApp : int): String Operasi ini digunakan untuk mengambil detail aplikasi dari database.</p> <p>+ getTableAppForInventaris (): string Operasi ini digunakan untuk mengambil data table dari database.</p> <p>+ getDetailAppForInventaris (idApp : int): string Operasi ini digunakan untuk mengambil detail aplikasi dari database.</p> <p>+ getTableAppForLapSI (): String Operasi ini digunakan untuk mengambil data table dari database.</p> <p>+ getDetailAppForLapSI (idApp : int): string Operasi ini digunakan untuk mengambil detail aplikasi dari database.</p> <p>+ getProsesBisnisForLapSI (idPengajuanAPPBaru : int): string Operasi ini digunakan untuk mengambil data proses bisnis sesuai dengan idpengajuanAplikasiBaru dari database.</p> <p>+ getFiturForLapSI (idProsesBisnis : int): string</p>	

Operasi ini digunakan untuk mengambil data fitur dari database.

2.2.3.55 Spesific Design Class AplikasiDAO

AplikasiDAO	<<entity>>
+ insertDataAPP (newAplikasi : Aplikasi): bool Operasi ini digunakan untuk memasukkan data aplikasi baru ke dalam database.	
+ getDataAPPForLaporanProjectManagemen (idApp : int): string Operasi ini digunakan untuk mengambil data aplikasi sesuai dengan idApp untuk laporan project manajemen dari database.	
+ updateStatusAplikasi (idApp : int, idStatus : int, tahunPengguna : int): bool Operasi ini digunakan untuk mengubah status aplikasi dalam database.	
+ getDataTableAPPForPenambahanFitur (idUnit : int): bool Operasi ini digunakan untuk mengambil data aplikasi untuk penambahan fitur aplikasi sesuai dengan idUnit dari database.	

2.2.3.56 Spesific Design Class FormDAO

FormDAO	<<entity>>
+ assignFiturAsForm (newForm : Form): string Operasi ini digunakan untuk menambahkan form baru	

kedalam database.

+ getDataFormNotAssigned (idFitur : int): string

Operasi ini digunakan untuk mengambil semua data form yang belum dikerjakan sesuai dengan idFitur dari database.

+ getDataAllForm (idFitur : int): string

Operasi ini digunakan untuk mengambil semua data form sesuai dengan idFitur dari database.

+ assignFormToStaff (updateForm : Form): bool

Operasi ini digunakan untuk mengubah form dengan menambahkan idStaff yang mengerjakan pada database.

+ assignFormToOutsource (updateForm : Form): bool

Operasi ini digunakan untuk mengubah form dengan menambahkan idOutsource yang mengerjakan pada database.

+ getDetailForm (idForm : int): string

Operasi ini digunakan untuk mengambil data detail form dalam database.

+ updateStatusToFinish (idForm : int, tanggalSelesai : string, idStatus : int): bool

Operasi ini digunakan untuk mengubah status form didatabase.

+ updateStatusToProgress (idForm : int, tanggalMulai : string, idStatus : int): bool.

Operasi ini digunakan untuk mengubah status form didatabase.

+ getFormTaskForStaff (idStaff : int, idStatus : int): string

Operasi ini digunakan untuk mengambil data form yang dikerjakan oleh staff tertentu dalam database.

+ getCompleteFormTaskForStaff (idStaff : int, idStatus

:int): string

Operasi ini digunakan untuk mengambil data form yang sudah selesai dikerjakan oleh staff tertentu dalam database.

2.2.3.57 Spesific Design Class ReportDAO

ReportDAO	<<entity>>
+ assignFiturAsReport (newReport : Report): string Operasi ini digunakan untuk menambahkan report baru kedalam database.	
+ getDataReportNotAssigned (idFitur : int): string Operasi ini digunakan untuk mengambil semua data report yang belum dikerjakan sesuai dengan idFitur dari database.	
+ getDataAllReport (idFitur : int): string Operasi ini digunakan untuk mengambil semua data report sesuai dengan idFitur dari database.	
+ assignReportToStaff (updateReport : Report): bool Operasi ini digunakan untuk mengubah report dengan menambahkan idStaff yang mengerjakan pada database.	
+ assignReportToOutsource (updateReport : Report): bool Operasi ini digunakan untuk mengubah report dengan menambahkan idOutsource yang mengerjakan pada database.	
+ getDetailReport (idReport : int): string Operasi ini digunakan untuk mengambil data detail report dalam database.	
+ updateStatusToFinish (idReport : int, tanggalSelesai : string, idStatus : int): bool Operasi ini digunakan untuk mengubah status report	

didatabase.

+ updateStatusToProgress (idReport : int, tanggalMulai : string, idStatus : int): bool.

Operasi ini digunakan untuk mengubah status report didatabase.

+ getReportTaskForStaff (idStaff : int, idStatus : int): string

Operasi ini digunakan untuk mengambil data report yang dikerjakan oleh staff tertentu dalam database.

+ getCompleteReportTaskForStaff (idStaff : int, idStatus : int): string

Operasi ini digunakan untuk mengambil data report yang sudah selesai dikerjakan oleh staff tertentu dalam database.

2.2.3.58 Spesific Design Class DesainDAO

DesainDAO	<<entity>>
+ assignFiturAsDesain (newDesain : Desain): string Operasi ini digunakan untuk menambahkan desain baru kedalam database.	
+ getDataDesainNotAssigned (idFitur : int): string Operasi ini digunakan untuk mengambil semua data desain yang belum dikerjakan sesuai dengan idFitur dari database.	
+ getDataAllDesain (idFitur : int): string Operasi ini digunakan untuk mengambil semua data desain sesuai dengan idFitur dari database.	
+ assignDesainToStaff (updateDesain : Desain): bool Operasi ini digunakan untuk mengubah desain dengan	

menambahkan idStaff yang mengerjakan pada database.

+ assignDesainToOutsource(updateDesain : Desain): bool

Operasi ini digunakan untuk mengubah desain dengan menambahkan idOutsource yang mengerjakan pada database.

+ getDetailDesain (idDesain : int): string

Operasi ini digunakan untuk mengambil data detail desain dalam database.

+ updateStatusToFinish (idDesain : int, tanggalSelesai : string, idStatus : int): bool

Operasi ini digunakan untuk mengubah status desain didatabase.

+ updateStatusToProgress (idDesain : int, tanggalMulai : string, idStatus : int): bool.

Operasi ini digunakan untuk mengubah status desain didatabase.

+ getDesainTaskForStaff (idStaff : int,idStatus :int): string

Operasi ini digunakan untuk mengambil data desain yang dikerjakan oleh staff tertentu dalam database.

+ getCompleteDesainTaskForStaff (idStaff : int,idStatus :int): string

Operasi ini digunakan untuk mengambil data desain yang sudah selesai dikerjakan oleh staff tertentu dalam database.

2.2.3.59 Spesific Design Class StaffDAO

StaffDAO	<<entity>>
+ getAllStaff (): string	
Operasi ini digunakan untuk mengambil semua data staff	

Program Studi Teknik Informatika	DPPL – SIKSI	75/ 100
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

yang ada dalam database.

+ getAllKSIStaff (): string

Operasi ini digunakan untuk mengambil semua data staff yang ada berada dalam unit KSI dalam database.

+ getStaffEmail (idStaff : int): string

Operasi ini digunakan untuk mengambil data email staff dalam database.

2.2.3.60 Spesific Design Class ResourceDAO

ResourceDAO	<<entity>>
+ getDataResource (): string Operasi ini digunakan untuk mengambil semua data pekerjaan yang sedang dilakukan staff dalam database.	
+ getTableAvailableStaff (): string Operasi ini digunakan untuk mengambil semua data staff yang ada tersedia dalam database.	

2.2.3.61 Spesific Design Class PrintLaporanManajerDAO

PrintLaporanManajerDAO	<<entity>>
+ getAllStaffSummary (tanggalSelesaiFrom : string, tanggalSelesaiTo : string): Data Set Operasi ini digunakan untuk mengambil semua data staff untuk staff summary yang ada dalam database.	
+ getLaporanDetailAplikasi (tanggalSelesaiFrom : string, tanggalSelesaiTo : string): Data Set Operasi ini digunakan untuk mengambil data detail aplikasi untuk laporan detail aplikasi dalam database.	

2.2.3.62 Spesific Design Class cetakSuratCCPDAO

cetakSuratCCPDAO	<<entity>>
+ cetakSuratCCPDAO (idStaff : int, tanggalSelesaiFrom : string, tanggalSelesaiTo : string) : Data Set Operasi ini digunakan untuk mengambil data pekerjaan staff yang ada dalam database.	

2.2.3.63 Spesific Design Class OutsourcedAO

OutsourcedAO	<<entity>>
+ getAllOutsource () : string Operasi ini digunakan untuk mengambil semua data outsource yang ada dalam database.	

3. Perancangan data

3.1 Dekomposisi Data

3.1.1 Deskripsi Entitas Aplikasi

Nama	Tipe	Panjang	Keterangan
ID_APP	Integer	-	id aplikasi, <i>Primary key</i>
nama_APP	Varchar	75	Nama aplikasi
keterangan	Varchar	max	Keterangan aplikasi
is_terminated	bool	-	status aplikasi berakhir
ID_pengajuan_APP_baru	Integer	-	id pengajuan aplikasi, <i>foreign key</i>
ID_server	Integer	-	id server, <i>foreign key</i>
ID_status	Integer	-	id status, <i>foreign key</i>
ID_arsitektur_aplikasi	Integer	-	id arsitektur aplikasi, <i>foreign key</i>
ID_database_server	Integer	-	id database server, <i>foreign key</i>
ID_bahasa_pemrograman	Integer	20	id bahasa pemrograman, <i>foreign key</i>
lokasi_fisik_APP	Varchar	68	lokasi fisik aplikasi
tahun_penggunaan	Varchar	4	tahun penggunaan aplikasi
ID_jenis_APP	Integer	-	id jenis aplikasi, <i>foreign key</i>
tools_yang_digunakan	Varchar	max	tools yang

			digunakan dalam pengembangan aplikasi
sistem_operasi_yang_digunakan	Varchar	max	sistem operasi yang digunakan aplikasi

3.1.2 Deskripsi Entitas Desain

Nama	Tipe	Panjang	Keterangan
ID_desain	Integer	-	id desain, <i>Primary key</i>
nama_desain	Varchar	75	Nama desain
ID_fitur	Integer	-	id fitur, <i>foreign key</i>
keterangan	Varchar	max	keterangan desain
ID_staff	Integer	-	id staff, <i>foreign key</i>
ID_outsource	Integer	-	id outsource , <i>foreign key</i>
tanggal_mulai	Integer	-	tangaal mulai pengerjaan desain
tanggal_target_pengerjaan	datetime	-	target pengerjaan desain
tanggal_selesai	datetime	-	tanggal selesai pengerjaan desain
ID_status	Integer	-	id status, <i>foreign key</i>

3.1.3 Deskripsi Entitas Fitur

Nama	Tipe	Panjang	Keterangan
ID_fitur	Integer	-	id fitur, <i>Primary key</i>

nama_fitur	Varchar	89	Nama fitur
deskripsi_fitur	Varchar	max	deskripsi fitur
fungsi_fitur	Varchar	max	fungsi fitur
ID_APP	Integer	-	id aplikasi, <i>foreign key</i>
ID_status	Integer	-	id status, <i>foreign key</i>
ID_proses_bisnis	Integer	-	id proses bisnis, <i>foreign key</i>
ID_detail_penambahan_fitur_APP	Integer	-	id detail penambahan fitur, <i>foreign key</i>

3.1.4 Deskripsi Entitas Form

Nama	Tipe	Panjang	Keterangan
ID_form	Integer	-	id form, <i>Primary key</i>
nama_form	Varchar	75	Nama form
ID_fitur	Integer	-	id fitur, <i>foreign key</i>
deskripsi_form	Varchar	max	deskripsi form
fungsionalitas_form	Varchar	max	fungsionalitas form
keterangan	Varchar	max	keterangan form
ID_staff	Integer	-	id staff, <i>foreign key</i>
ID_outsource	Integer	-	id outsource , <i>foreign key</i>
tanggal_mulai	Integer	-	tangaal mulai pengerjaan form
tanggal_target_pengerjaan	datetime	-	target pengerjaan form
tanggal_selesai	datetime	-	tanggal selesai

Program Studi Teknik Informatika	DPPL – SIKSI	80/ 100
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

			pengerjaan form
ID_status	Integer	-	id status, <i>foreign key</i>
ID_model_fitur	Integer	-	id model fitur, <i>foreign key</i>

3.1.5 Deskripsi Entitas Outsource

Nama	Tipe	Panjang	Keterangan
ID_outsource	Integer	-	id outsource, <i>Primary key</i>
nama_outsource	Varchar	80	Nama outsource
alamat_outsource	Varchar	80	id fitur, <i>foreign key</i>
no_telepon_outsource	Varchar	80	deskripsi form

3.1.6 Deskripsi Entitas Report

Nama	Tipe	Panjang	Keterangan
ID_report	Integer	-	id report, <i>Primary key</i>
nama_report	Varchar	75	Nama report
ID_fitur	Integer	-	id fitur, <i>foreign key</i>
deskripsi_report	Varchar	max	deskripsi report
keterangan	Varchar	max	keterangan report
ID_staff	Integer	-	id staff, <i>foreign key</i>
ID_outsource	Integer	-	id outsource , <i>foreign key</i>
tanggal_mulai	Integer	-	tangaal mulai pengerjaan report
tanggal_target_pengerjaan	datetime	-	target pengerjaan report
tanggal_selesai	datetime	-	tanggal selesai

			pengerjaan report
ID_status	Integer	-	id status, <i>foreign key</i>
ID_model_fitur	Integer	-	id model fitur, <i>foreign key</i>

3.1.7 Deskripsi entitas server

Nama	Tipe	Panjang	Keterangan
ID_server	Integer	-	id server, <i>Primary key</i>
nama_server	Varchar	50	Nama server
lokasi_server	Varchar	150	lokasi server
deskripsi_server	Varchar	max	deskripsi server
keterangan	Varchar	max	keterangan server
IP_server	Varchar	18	ip server

3.1.8 Deskripsi Entitas Staff

Nama	Tipe	Panjang	Keterangan
ID_staff	Integer	-	id staff, <i>Primary key</i>
nama_staff	Varchar	50	Nama staff
jabatan_staff	Varchar	50	jabatan staff
email	Varchar	30	email staf
alamat	Varchar	70	alamat staff
no_telp	Varchar	18	no telp staff
ID_unit	Integer	-	id unit, <i>foreign key</i>

3.1.9 Deskripsi Entitas Unit

Nama	Tipe	Panjang	Keterangan
ID_unit	Integer	-	id unit, <i>Primary key</i>
deskripsi_unit	Varchar	88	Nama staff
ID_staff	Integer	-	id staff, <i>foreign key</i>

3.1.10 Deskripsi Entitas Arsitektur Aplikasi

Nama	Tipe	Panjang	Keterangan
ID_arsitektur_aplikasi	Integer	-	id arsitektur aplikasi, <i>Primary key</i>
deskripsi_arsitektur_aplikasi	Varchar	40	deskripsi arsitektur aplikasi

3.1.11 Deskripsi Entitas Bahasa Pemrograman

Nama	Tipe	Panjang	Keterangan
ID_bahasa_pemrograman	Integer	-	id bahasa pemrograman, <i>Primary key</i>
deskripsi_bahasa_pemrograman	Varchar	48	deskripsi bahasa pemrograman

3.1.12 Deskripsi Entitas Database Server

Nama	Tipe	Panjang	Keterangan
ID_database_server	Integer	-	id database server, <i>Primary key</i>
deskripsi_database_server	Varchar	50	deskripsi database server

3.1.13 Deskripsi Entitas Jenis Aplikasi

Nama	Tipe	Panjang	Keterangan
ID_jenis_APP	Integer	-	id jenis aplikasi, <i>Primary key</i>
deskripsi_jenis_APP	Varchar	68	deskripsi jenis aplikasi

3.1.14 Deskripsi Entitas Model Fitur

Nama	Tipe	Panjang	Keterangan
ID_model_fitur	Integer	-	id model fitur, <i>Primary key</i>

deskripsi_model_fitur	Varchar	88	deskripsi model fitur
kompleksitas	Integer	-	kompleksitas model
model_kode	Varchar	18	kode model

3.1.15 Deskripsi Entitas Privilege

Nama	Tipe	Panjang	Keterangan
ID_privilege	Integer	-	id privilege, <i>Primary key</i>
deskripsi_privilege	Varchar	38	deskripsi privilege

3.1.16 Deskripsi Entitas Sistem Operasi

Nama	Tipe	Panjang	Keterangan
ID_sistem_operasi	Integer	-	id sistem operasi, <i>Primary key</i>
deskripsi_sistem_operasi	Varchar	38	deskripsi sistem operasi

3.1.17 Deskripsi Entitas Bahasa Pemrograman

Nama	Tipe	Panjang	Keterangan
ID_status	Integer	-	id status, <i>Primary key</i>
deskripsi_status	Varchar	38	deskripsi status

3.1.18 Deskripsi Entitas Tools Pengembangan

Nama	Tipe	Panjang	Keterangan
ID_tools_pengembangan	Integer	-	id tools pengembangan, <i>Primary key</i>
deskripsi_tools_pengembangan	Varchar	78	deskripsi tools pengembangan

3.1.19 Deskripsi Entitas Pengajuan Aplikasi baru

Nama	Tipe	Panjang	Keterangan
ID_pengajuan_APP_baru	Integer	-	id pengajuan aplikasi, <i>Primary key</i>
deskripsi_APP	Varchar	max	deskripsi pengajuan aplikasi
tanggal	datetime	-	tanggal pengajuan
ID_unit	Integer	-	id unit, <i>foreign key</i>
ID_staff	Integer	-	id staff, <i>foreign key</i>
fungsi_tujuan_APP	Varchar	max	fungsi dan tujuan aplikasi yang diajukan
target_waktu_operasional	datetime	-	target waktu operasional aplikasi
keterangan	Varchar	max	keterangan pengajuan aplikasi
ID_status	Integer	-	id status, <i>foreign key</i>
pengguna_APP	Varchar	max	pengguna aplikasi

3.1.20 Deskripsi Entitas Permintaan Penambahan Fitur Aplikasi

Nama	Tipe	Panjang	Keterangan
ID_permintaan_penambahan_fitur_A PP	Integer	-	id permintaan penambahan fitur, <i>Primary</i>

			<i>key</i>
ID_unit	Integer	-	id unit, <i>foreign</i> <i>key</i>
ID_APP	Integer	-	id app, <i>foreign</i> <i>key</i>
tanggal_pengajuan_permintaan	Datetime	-	tanggal pengajuan permintaan penambahan

3.1.21 Deskripsi Entitas Users

Nama	Tipe	Panjang	Keterangan
user_name	Varchar	50	id permintaan penambahan fitur, <i>Primary</i> <i>key</i>
privilege	Varchar	max	id unit, <i>foreign</i> <i>key</i>
ID_staff	Integer	-	id app, <i>foreign</i> <i>key</i>
password	Varchar	max	tanggal pengajuan permintaan penambahan
Is_deleted	bool	-	status use dihapus

3.1.22 Deskripsi Entitas Detail Penambahan Fitur

Nama	Tipe	Panjang	Keterangan
nama_fitur	Varchar	200	Nama fitur

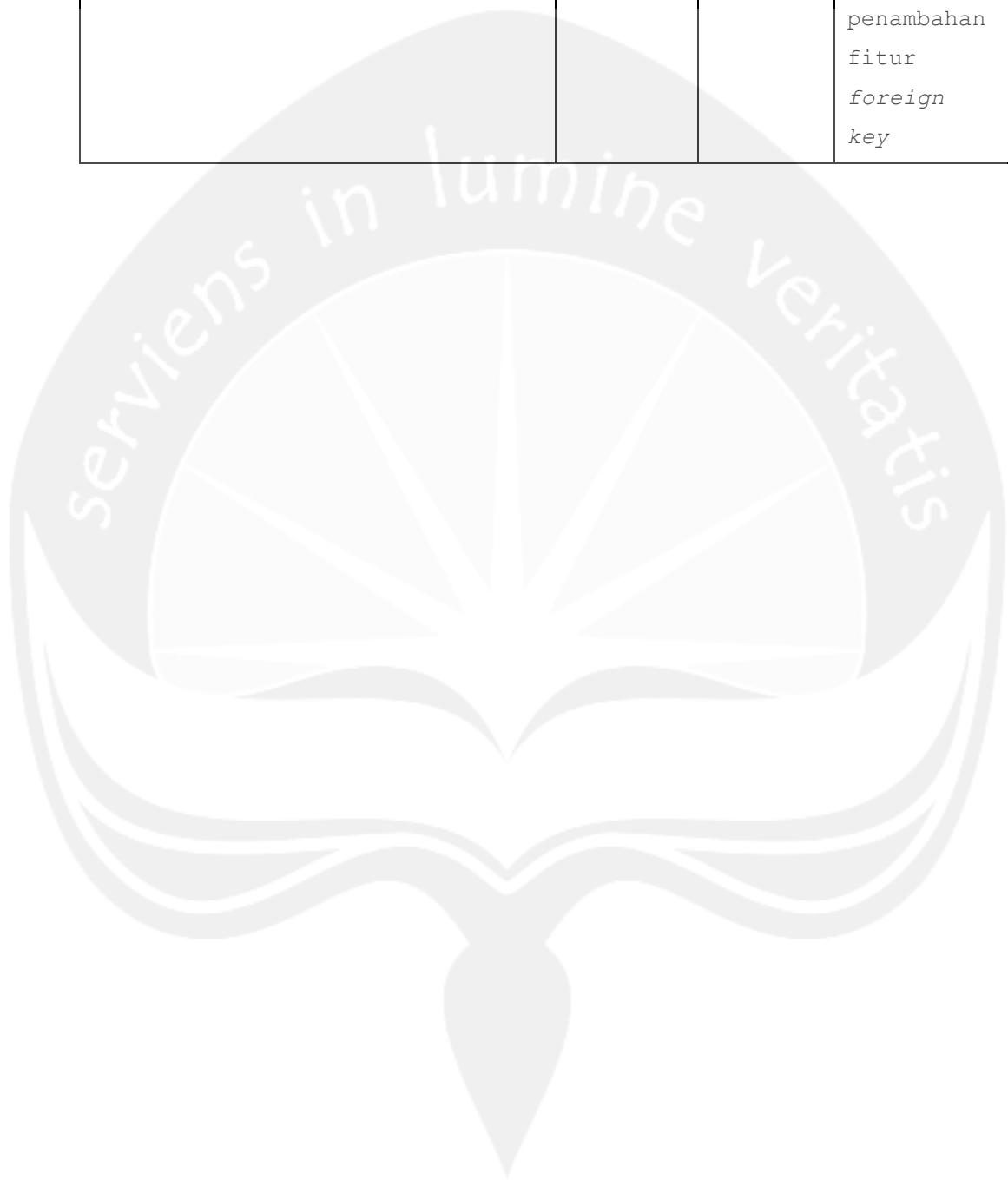
Program Studi Teknik Informatika	DPPL – SIKSI	86/ 100
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

deskripsi_fitur	Integer	max	deskripsi fitur
latar_belakang	Varchar	max	latar belakang
ID_permintaan_penambahan_fitur_APP	Integer	-	id permintaan penambahan fitur <i>foreign key</i>
ID_status	Integer	-	id status <i>foreign key</i>
ID_detail_penambahan_fitur_APP	Integer	-	id detail penambahan fitur <i>primary key</i>
ID_proses_bisnis	Integer	-	id proses bisnis <i>foreign key</i>

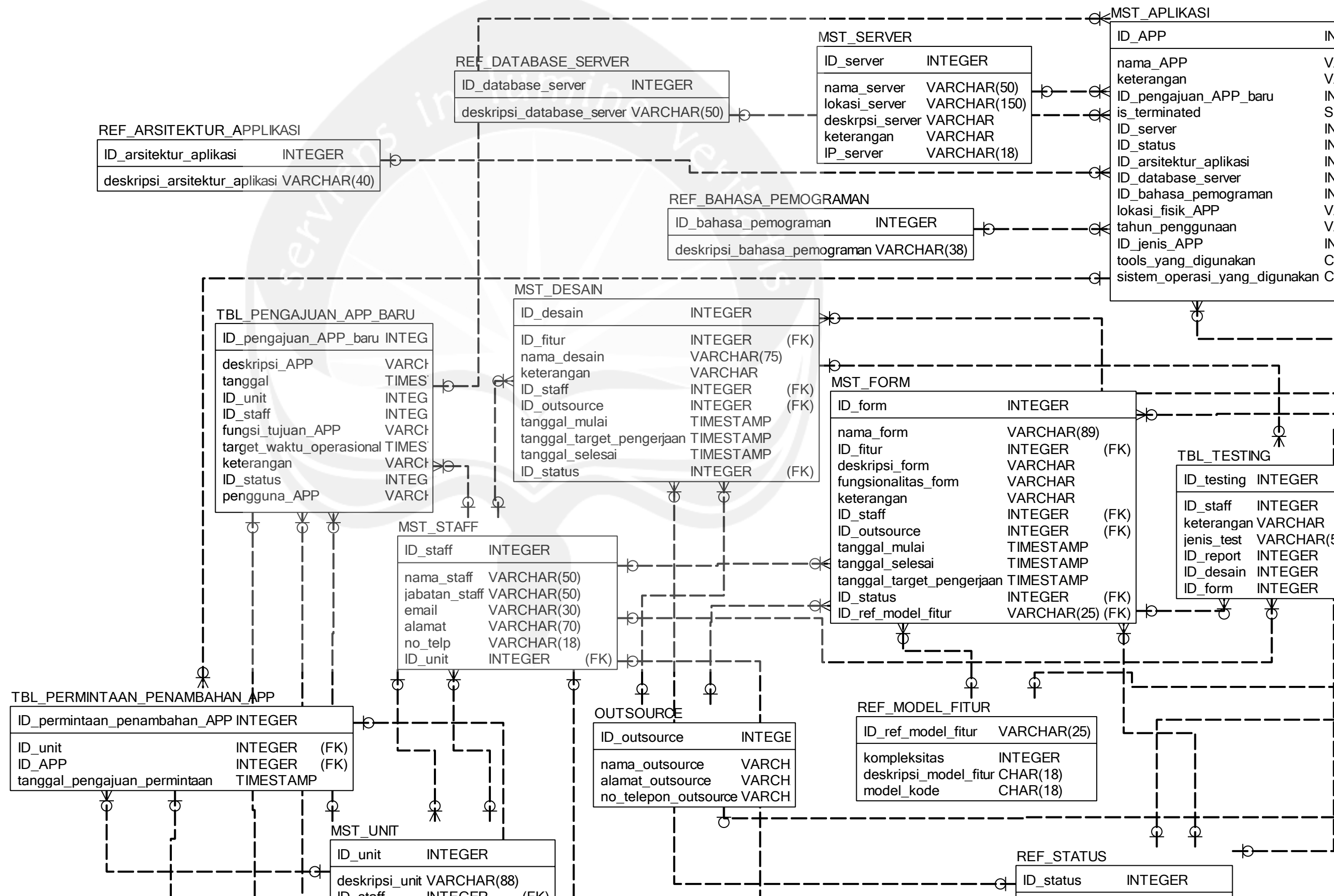
3.1.23 Deskripsi Entitas Detail Proses Bisnis

Nama	Tipe	Panjang	Keterangan
ID_proses_bisnis	Integer	-	id proses bisnis <i>primary key</i>
deskripsi_proses_bisnis	Varchar	100	deskripsi proses bisnis
ID_pengajuan_APP_baru	Integer	-	id pengajuan app <i>foreign</i>

			<i>key</i>
ID_permintaan_penambahan_fitur_A PP	Integer	-	id permintaan penambahan fitur <i>foreign</i> <i>key</i>

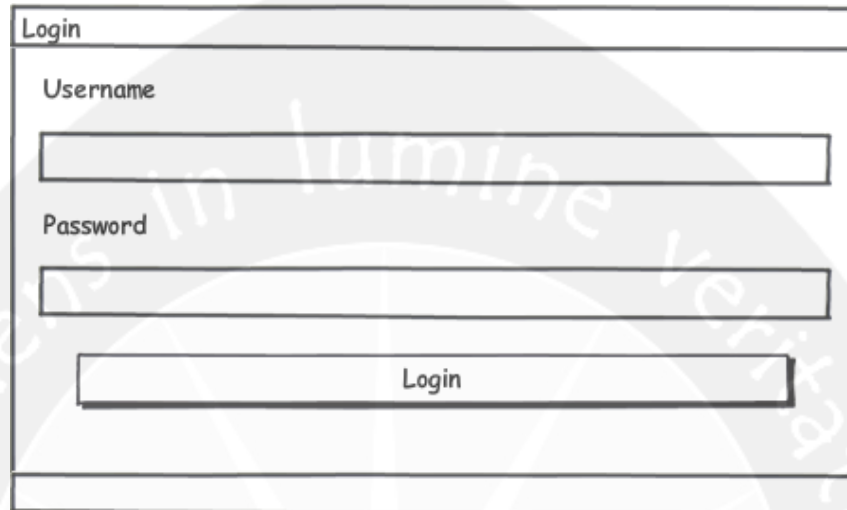


3.1 Physic Data Model



4. Perancangan Antarmuka

4.1 Login



The diagram shows a rectangular window titled "Login". Inside the window, there are two text input fields. The first field is preceded by the label "Username" and the second field is preceded by the label "Password". Below these two fields is a button with the text "Login".

Gambar 31. Rancangan Antarmuka Login

Antarmuka ini digunakan sebagai antarmuka untuk login. Pengguna akan membuka page ini pertama kali masuk ke dalam sistem. Yang perlu diinputkan adalah username dan password. Jika login berhasil, maka akan muncul akan masuk ke halaman index. Jika login tidak berhasil (username atau password yang diinputkan tidak sesuai dengan data yang ada pada basis data) akan muncul pesan kesalahan.

4.2 Laporan Sistem Informasi

No	Nama	Tanggal	Tanggal Target	Status
-	-	-	-	-

Gambar 32. Rancangan Antarmuka Laporan Sistem Informasi

Antarmuka ini digunakan untuk menampilkan laporan sistem informasi. Atribut yang ditampilkan dalam laporan ini berupa no, nama aplikasi, tanggal pengajuan, tanggal target operasional dan status. Pada antarmuka ini pengguna bisa mencari aplikasi yang telah dibuat.

4.3 Inventaris Sistem Informasi

No	Nama	Tanggal Pengajuan	Tanggal Operasional	Status Aplikasi	Deploy Aplikasi	Detail
1	-	-	-	-	-	Detail
2	-	-	-	-	-	Detail
3	-	-	-	-	-	Detail

Gambar 33. Rancangan Laporan Sistem informasi

Antarmuka ini digunakan untuk menampilkan inventarisasi sistem informasi. Pada antarmuka ini atribut yang ditampilkan nama aplikasi, tanggal pengajuan, tanggal operasional, status aplikasi ,

deploy aplikasi dan detail. Pada kolom detail terdapat tombol detail, yang menampilkan detail dari aplikasi.

4.4 Detail Inventaris Sistem Informasi

Inventaris Sistem Informasi							
Tabel Aplikasi							
Navigation	No	Nama	Tanggal Pengajuan	Tanggal Operasional	Status Aplikasi	Deploy Aplikasi	Detail
	1	-	-	-	-	-	Detail
	2	-	-	-	-	-	Detail
	3	-	-	-	-	-	Detail

Gambar 34. Rancangan Antarmuka Detail Inventaris Sistem Informasi

Antarmuka ini digunakan untuk menampilkan inventaris aplikasi. Pada antarmuka ini atribut yang akan ditampilkan berupa nama aplikasi, deskripsi aplikasi, fungsi dan tujuan, keterangan aplikasi, keterangan pengajuan aplikasi, pemilik aplikasi, lokasi fisik aplikasi, tools yang digunakan, bahasa yang digunakan, arsitektur aplikasi, database, sistem operasi, dan server yang digunakan.

4.5 Pengajuan Permohonan Pembuatan Aplikasi Baru

Pengajuan Pembuatan Aplikasi Baru

Navigation

Aplikasi Yang Dimiliki

No	Nama	Tanggal Pengajuan	Tanggal Target Operasional	Status
1	-	-	-	-
2	-	-	-	-

Data Pengajuan Aplikasi

No	Tanggal Pengajuan	Staff	Status
1	-	-	-
2	-	-	-

Form Pengajuan Permohonan

Gambar 35. Rancangan Antarmuka Pengajuan Permohonan Pembuatan Aplikasi Baru

Antarmuka ini digunakan untuk mengajukan permohonan pembuatan aplikasi baru. Pada antarmuka ini terdapat dua group field yaitu aplikasi yang dimiliki dan data pengajuan aplikasi. Pada group field aplikasi menampilkan atribut berupa no, nama aplikasi, tanggal pengajuan, tanggal target operasional dan status, sedangkan untuk group field data pengajuan aplikasi menampilkan atribut no, tanggal pengajuan staff dan status. Pada bagian bawah terdapat tombol untuk melakukan pengajuan permohonan yang akan masuk ke halaman pengajuan pembuatan aplikasi baru.

4.6 Form Pengajuan Pembuatan Aplikasi Baru

Pengajuan Pembuatan Aplikasi Baru	
Nama Unit(Biro/Fakultas/Lembaga/Lainya) :	-
Penanggung Jawab/Contact Person Unit :	-
Jabatan Penanggung Jawab :	-
Telpn (Ekstensi) :	-
Email Penanggung Jawab :	-
Waktu Target Operasional Software (Bulan/Tahun) :	-
Fungsi.Tujuan (Singkat) Software :	-
Deskripsi Singkat Aplikasi :	-
Pengguna Aplikasi :	-
Proses Bisnis :	-
Keterangan :	-

Gambar 36. Rancangan Antarmuka Form Pengajuan Pembuatan Aplikasi Baru

Antarmuka ini digunakan untuk melakukan input pembuatan aplikasi baru. Pada antarmuka ini pengguna memasukkan inputan yang diperlukan sebagai detail permohonan pengajuan pembuatan aplikasi baru. Atribut yang akan dimasukan berupa nama unit, penanggung jawab, jabatan penganggung jawab, nomor telepon, email penanggung jawab, waktu target operasional software (dalam bulan / tahun), fungsi dan tujuan singkat software, deskripsi singkat software, pengguna aplikasi, proses bisnis, dan keterangan.

4.7 Pengajuan Penambahan Fitur

N	Nama	Tanggal	Tanggal Target	Statu
1	-	-	-	-
2	-	-	-	-

N	Nama	Deskripsi	Latar	Status	
1	-	-	-	-	-
2	-	-	-	-	-

Gambar 37. Rancangan Antarmuka Pengajuan Penambahan Fitur

Antarmuka ini digunakan untuk pengajuan penambahan fitur. Pada antarmuka ini terdapat dua group field yaitu, group field aplikasi dan data pengajuan fitur. Pada group field table aplikasi, akan ditampilkan atribut no, nama aplikasi, tanggal pengajuan, tanggal target operasional dan status. Pada group field data pengajuan fitur akan ditampilkan atribut no, nama fitur, deskripsi fitur, latar belakang, status pengajuan dan tanggal pengajuan.

4.8 Laporan Perkembangan Project

- + Sistem Informasi Assets
- + Pendataan Asset
- + Pelaporan Asset
- + Pengujian Asset

Gambar 38. Rancangan Antarmuka Laporan Perkembangan Project

Antarmuka ini digunakan untuk menampilkan perkembangan project yang telah dilakukan pada antarmuka ini pengguna bisa melihat detail perkembangan pembangunan aplikasi.

4.9 Project Management

The screenshot shows a web application interface for Project Management. It features a sidebar with a 'Navigation' menu. The main content area is divided into several sections:

- Project Management**: A header section.
- Permohonan Aplikasi Yang Diajukan**: A table with columns: Deskripsi Aplikasi, Tanggal Pengajuan, Unit, Fungsi Aplikasi, Target Operasional, and Approve. The table contains one row with dashes in all cells.
- Kelola Aplikasi**: A section with a search bar labeled 'Cari Aplikasi' and a search button.
- Sistem Informasi**: A table with columns: Nama Aplikasi and Keterangan. The table contains one row with dashes in both cells.
- Proses Bisnis**: A table with columns: No and Deskripsi Proses. The table contains two rows: '1 Pendataan Asset' and '2 Pelaporan Asset'.
- Fitur Aplikasi**: A table with columns: Nama, Deskripsi, and Fitur. The table contains one row with dashes in all three cells.
- add fitur**: A button at the bottom of the Fitur Aplikasi section.

Gambar 39. Rancangan Antarmuka Project Management

Antarmuka ini digunakan untuk melakukan manajemen project. Pada antarmuka ini terdapat empat group field yaitu permohonan aplikasi yang diajukan, kelola aplikasi, sistem informasi, proses bisnis, fitur aplikasi. Pada group field permohonan aplikasi yang diajukan akan ditampilkan atribut deskripsi aplikasi, tanggal pengajuan, unit, fungsi aplikasi, target operasional, approve. Pada group field kelola aplikasi, pengguna dapat mencari aplikasi yang telah diajukan. Pada group field proses bisnis, pengguna dapat melihat proses bisnis yang telah diajukan berupa atribut no dan

deskripsi proses. Pada group field fitur aplikasi pengguna dapat melihat fitur aplikasi yang telah ada berupa atribut nama, deskripsi dan fitur. Pada bagian bawah terdapat tombol add fitur yang dapat digunakan untuk menginputkan fitur secara lebih mendetail.

4.10 Penugasan Project – Penugasan Project

The screenshot shows a web application interface titled "Penugasan Project". On the left is a "Navigation" pane. On the right, there are three stacked tables, each with a title bar and four columns: "Nama", "Tanggal Penugasan", "Target Pengerjaan", and "Status".

Penugasan Form			
Nama Form	Tanggal Penugasan	Target Pengerjaan	Status
-	-	-	-

Penugasan Report			
Nama Report	Tanggal Penugasan	Target Pengerjaan	Status
-	-	-	-

Penugasan Desain			
Nama Desain	Tanggal Penugasan	Target Pengerjaan	Status
-	-	-	-

Gambar 40. Rancangan Antarmuka Penugasan Project

Antarmuka ini digunakan untuk menampilkan penugasan project. Pada antarmuka ini terdapat tiga group field yaitu penugasan form, penugasan report, penugasan desain. Pada group field penugasan form akan menampilkan atribut nama form, tanggal penugasa, target pengerjaan dan status. Pada group field penugasan report akan menampilkan atribut nama form, tanggal penugasa, target pengerjaan dan status. Pada group field penugasan desain akan menampilkan atribut nama form, tanggal penugasa, target pengerjaan dan status.

4.11 Task - On Progress Task

The wireframe shows a window titled 'Task'. On the left is a 'Navigation' pane. On the right, there are two tabs: 'On Progress Task' (active) and 'Completed Task'. Under the 'On Progress Task' tab, there are two sections: 'Form Task' and 'Report Task'. Each section contains a table with four columns: 'Nama Form', 'Tanggal Mulai Pengerjaan', 'Target Pengerjaan', and 'Status'. Each table has one row with hyphens as placeholders.

Task			
On Progress Task Completed Task			
Form Task			
Nama Form	Tanggal Mulai Pengerjaan	Target Pengerjaan	Status
-	-	-	-

Report Task			
Nama Form	Tanggal Mulai Pengerjaan	Target Pengerjaan	Status
-	-	-	-

Gambar 41. Rancangan Antarmuka Task - On Progress Task

Antarmuka ini digunakan untuk untuk menampilkan task yang sedang berjalan dan yang sudah selesai. Terdapat dua tab bar pada antarmuka ini yaitu On Progress Task dan Completed Task. Pada tab bar On Progress Task terdapat dua group field yaitu Form Task dan Report Task. Pada group field on progress task, akan menampilkan nama form, tanggal mulai pengerjaan, target pengerjaan, status.

4.12 Task - Complete Task

The screenshot shows a web application interface titled 'Task'. It has a 'Navigation' sidebar on the left. The main content area has two tabs: 'On Progress Task' and 'Completed Task'. The 'Completed Task' tab is active, displaying three task lists:

- Form Task**

Nama	Tanggal Mulai	Target	Tanggal
-	-	-	-
- Report Task**

Nama	Tanggal Mulai	Target	Tanggal
-	-	-	-
- Desain Task**

Nama Form	Tanggal Mulai	Target	Status
-	-	-	-

Gambar 42. Rancangan Antarmuka Task - Complete Task

Antarmuka ini digunakan untuk menampilkan Completed Task. Pada antarmuka ini terdapat dua group field, yaitu report task dan desain task. Pada group field report task akan ditampilkan atribut nama form, tanggal mulai pengerjaan, target pengerjaan, tanggal selesai dan status. Pada group field desain task akan ditampilkan atribut nama form, tanggal mulai pengerjaan, target pengerjaan dan status.

4.13 Cetak Surat CPP

No	Nama	Tanggal	Tanggal Target	Status
-	-	-	-	-

Pada antarmuka ini, pengguna dapat mencetak surat CPP. Untuk mencetak surat ccp dapat dilakukan dengan menginputkan tanggal dari dan sampai, setelah itu tekan tombol Cetak Surat CCP.